BOEING

International Space Station Program

D684-10017-01

Software Development Plan

Rev B

Type 2 Approved by NASA

June 30, 2000

Submitted to: National Aeronautics and Space Administration
Johnson Space Center
Contract No. NAS 15-10000 (DR VE-29)

REVISION AND HISTORY PAGE

REV.	DESCRIPTION	PUB. DATE
-	Initial Release	04-08-94
A	Revision A incorporates ECP 217	06-03-97
	DCN 001 incorporates CCR 840	05-27-98
	DCN 002 incorporates CCR 313, 1560, 1620	09-27-99
	DCN 003 incorporates CCR 1720	09-27-99
B	Revision B per incorporates SSCN 002602 and all previous DCNs. Changes the name of the document from "Prime Contractor Software Development Plan" to "Software Development Plan."	10-13-00

ERU: /s/ Mary C. Nooney 10-13-00

This page left blank intentionally.

INTERNATIONAL SPACE STATION PROGRAM

SOFTWARE DEVELOPMENT PLAN

REV B

PREFACE

This Boeing document D684-10017-1, Space Station Software Development Plan (SDP), describes the Boeing approach for effectively documenting, controlling, developing and testing software for the United States (U.S.) segments of the International Space Station (ISS) Program. This plan is prepared for NASA approval in accordance with Space Station Prime Contract Data Requirement (DR) VE-29 and is maintained by the Command and Data Handling (C&DH) Software Integration (SWI) Integrated Product Team (IPT). Information provided herein is submitted in accordance with the guidelines of DOD-STD-2167A. Development of software by Tier 1 Subcontractors will be in consonance with this plan and will be documented in their respective SDPs.

KEY WORDS

Command and Data Handling
Configuration Control
Configuration Management
DOD-STD-2167
End Item
Firmware Controller
Formal Qualification Testing
Independent Verification and Validation
Integrated Product Team
International Space Station
Mission Build Facility
Multiplexer/Demultiplexer
Prototype Software
Software Development Plan
Software Engineering Environment
Software Metrics
Software Verification Facility
Tier 1 Subcontractor

/s/W. Panter	9/14/00	/s/C. J. Howard	9/14/00
NASA Co-Chairman Avionics and Software Manager	Date	Boeing Co-Chairman Avionics and Software Manager	Date

CONCURRENCE

INTERNATIONAL SPACE STATION PROGRAM

SOFTWARE DEVELOPMENT PLAN

REV B

PREPARED BY:	GEORGE WIDERQUIST PRINT NAME	GEOCONTROL SYSTEMS ORG
	/s/George R. Widerquist SIGNATURE	29 Aug 2000 DATE
CHECKED BY:	SCOT WILKE PRINT NAME	BOEING SW INTEGRATION ORG
	/s/Scot N. Wilke SIGNATURE	9/13/00 DATE
SUPERVISED BY:	for JIM SIMPFENDERFER PRINT NAME	BOEING SOFTWARE ORG
	/s/J. W. Sherrill SIGNATURE	9/13/00 DATE
SUPERVISED BY:	for PEGGY THOMAS PRINT NAME	BOEING ASQPI ORG
	/s/Robert D. Whiting SIGNATURE	9/13/2000 DATE
SUPERVISED BY:	DAVID PRUETT PRINT NAME	NASA AVIONICS & SOFTWARE ORG
	/s/David Pruett SIGNATURE	9/14/00 DATE
DQA:	HOPE KILPATRICK PRINT NAME	BOEING DIT ORG
	/s/Hope Kilpatrick SIGNATURE	9/14/00 DATE

INTERNATIONAL SPACE STATION PROGRAM

SOFTWARE DEVELOPMENT PLAN

LIST OF CHANGES

REV B

All changes to paragraphs, tables, and figures in this document are shown below:

ENTRY DATE	CHANGE	PARAGRAPH(S)
30 Jun 2000	Document title and revision	n/a
30 Jun 2000	Identification data	1.1
30 Jun 2000	System Overview	1.2
30 Jun 2000	Document Overview	1.3
30 Jun 2000	Relationship to other plans	1.4
30 Jun 2000	Reference added	2
30 Jun 2000	B-HB replaces MDA	3.1.3.2
30 Jun 2000	PCS office redefined	3.1.3.6
30 Jun 2000	Removal of unneeded paragraphs	3.8
30 Jun 2000	Realignment of reviews held for NASA	3.8.2
30 Jun 2000	NASA defined as customer	3.8.2.1
30 Jun 2000	Added D684-10097-01 as governing FCA document	3.8.2.5
30 Jun 2000	Added D684-10097-01 as governing PCA document	3.8.2.6
30 Jun 2000	Stage Integration rewrite	3.8.3
30 Jun 2000	Renumber of section from 3.8.3.4 to 3.8.3.1; Modification of inputs and success criteria; Deletion of sections 3.8.3.1, 3.8.3.2, 3.8.3.3, 3.8.3.5	3.8.3.1
30 Jun 2000	Boeing site name changes	3.15
30 Jun 2000	Boeing site name changes; restricted mandating to only products used for production of flight software	4.1.3
30 Jun 2000	Addition of PPL maturity level definitions	4.2.3.3.3
30 Jun 2000	Rework of Patch/PPL process; removal of forms	8.1
30 Jun 2000	Introduction of Sustaining process	8.2ff
30 Jun 2000	Modification of acronyms	9.1

LIST OF CHANGES (CONTINUED)

ENTRY DATE	CHANGE	FIGURE(S)
30 Jun 2000	Redescribed informal delivery	8.1-1
30 Jun 2000	Deleted	8.1-2
30 Jun 2000	Deleted	8.1-3
30 Jun 2000	Addition of Sustaining Process	8.2-1

ENTRY DATE	CHANGE	TABLES(S)
30 Jun 2000	Addition	4.2.3.3.3-1
30 Jun 2000	Reduction in content	8.1-1
30 Jun 2000	Reduction in content	8.1-2
30 Jun 2000	Reduction in content	8.1-3

ENTRY DATE	CHANGE	APPENDICES(S)
30 Jun 2000	Boeing site name changes	Table A-1
30 Jun 2000	Boeing site name changes; addition of HCOR, MDMLU, SSMMU	Table A-2
30 Jun 2000	Boeing site name changes; addition of ActivAda PR Compiler	Table A-3
30 Jun 2000	Boeing site name changes	Table A-4
30 Jun 2000	Boeing site name changes	Table A-5
30 Jun 2000	Boeing site name changes; addition of ActivAda PR Compiler	Table C.1-1

TABLE OF CONTENTS

TABLE OF CONTENTS

PARAGRAPH **PAGE**

TABLE OF CONTENTS

PARAGRAPH **PAGE**

TABLE OF CONTENTS

PARAGRAPH **PAGE**

TABLE OF CONTENTS

PARAGRAPH **PAGE**

TABLE OF CONTENTS

PARAGRAPH **PAGE**

APPENDICES

APPENDIX **PAGE**

APPENDICES

APPENDIX **PAGE**

LIST OF FIGURES

NUMBER **PAGE**

LIST OF FIGURES

NUMBER **PAGE**

TABLES

NUMBER **PAGE**

This page left blank intentionally.

1 SCOPE

1.1 IDENTIFICATION

This Software Development Plan is provided as Data Requirements List (DRL) item VE-29 for the Boeing software development effort for the International Space Station Program. This document establishes the policies, procedures and guidelines for the development and test of the following types of software, either developed or acquired for the U.S. segments of the ISS: Rev B

A. Flight software;

B. Ground software (including Mission Build Facility (MBF) software);

C. Ground Support Equipment (GSE) software and Test Support Equipment (TSE) software;

D. Test software, including simulations; and

E. Software Verification Facility (SVF) software.

This SDP does not govern the following ISS software:

A. Software developed by the International Partners (IPs), including items C through G in Section 1.2;

B. Software contracted by the National Aeronautics and Space Administration (NASA) outside of the Prime contract, including the Control Center Complex (CCC), Test Control and Monitor System (TCMS) at Kennedy Space Center (KSC) and training unique software and training facility developed at Johnson Space Center (JSC);

C. Payload/User software;

D. Government Furnished Equipment (GFE), including Timeliner Kernels and Adapters, Orbiter Interface Unit (OIU), Columbus Ground Software (CGS) tools and Portable Computer System (PCS) software;

E. Cargo Planning Analysis and Configuration System developed by Boeing in Huntsville;

F. Functional Cargo Block (FGB) software developed by M.V. Khrunichev State Scientific Production Space Center (KhSC) in Moscow following SSP 50094, NASA/RSA Joint Specifications/Standards for the ISSA Russian Segment; and

G. Factory Equipment (FE) software.

The U.S. Space Station developmental software is partitioned into Computer Software Configuration Items (CSCIs) and will be developed and documented in accordance with DOD-STD-2167A, with appropriate tailoring as defined in the Software Standards and Procedures Specification (SSPS), D684-10056-1. Changes to the tailoring in the SSPS first require approval of the Boeing ISS Software Manager and then approval by NASA through the change control process. Rev B

1.2 SYSTEM OVERVIEW

The ISS system is comprised of an on-orbit facility, which is logistically supported with consumables, maintenance items, experiments, and ground facilities. Communications between the on-orbit ISS and ground facilities for command, control and status is provided by a satellite link.

The ISS is comprised of the following segments:

A. United States On-orbit Segment (USOS);

B. United States Ground Segment (USGS);

C. European Space Agency (ESA) Attached Pressurized Module (APM);

D. Japanese Experiment Module (JEM);

E. Russian Segment (RS);

F. Italian Mini-Pressurized Logistics Module (MPLM); and

G. Canadian Mobile Servicing System (MSS).

The USOS is the core of the orbiting facility and supplies resources for the other segments of the orbiting facility. The USGS is comprised of several facilities that are used to deploy, operate and maintain the orbiting facility. These facilities provide the capability to: process cargo, experiments and end items prior to launch; monitor and effect control of the orbiting facility; perform payload operations; resupply the orbiting facility with experiments, consumables and spares; provide maintenance support; and perform planning and provide simulation of the orbiting facility for verification and training.

The Boeing software development life-cycle is based on incremental delivery of functionality to satisfy staging requirements. Flight CSCI allocation will be one CSCI for each processor containing software with the exception of the Multiplexer/Demultiplexer (MDM) processors which all contain a Boot and Diagnostics firmware CSCI, a Serial/Parallel Data (SPD) Card 1553 firmware CSCI and a High Rate Data Link (HRDL) firmware CSCI in addition to a unique application CSCI. A complete list of the CSCIs covered by this SDP and the Boeing Software Development Groups (BSDGs) responsible for their development is included in Appendix A. A BSDG is also referred to in this Plan as a development group or Tier 1 Subcontractor. Rev B

Following the ISS redesign, some software products of the Space Station Freedom Program (SSFP) will be applicable to the new program. Section 3 provides the guidelines and criteria for “grandfathering” these products.

1.3 DOCUMENT OVERVIEW

This SDP describes the comprehensive plan for the management of the Space Station’s U.S. software development and provides the C&DH Subsystem Provider IPT with the means to define software development, life-cycle policies, procedures and guidelines; coordinate schedules; control computational resources at the processor level; and monitor progress of the subcontractor’s software development activity. This Plan also provides the BSDGs with the Rev B methods and processes to follow in performing their software development and defines the channels to the Program for software related communication. Rev B

This SDP meets or exceeds the requirements of DOD-STD-2167A, Data Item Description (DID) DI-MCCR-80030 with tailoring as identified in the Prime Contractor SSPS. Section 3, "Software Development Management", describes the planning associated with the software development management activities. Section 4, "Software Engineering," describes the techniques, methods and processes to be used in software development. Section 5, "Formal Qualification Testing", describes the planning associated with software formal testing activities. Section 6, "Software Product Evaluations," describes the planning associated with software product evaluation activities. Section 7, "Software Configuration Management", describes the software configuration management activities. Section 8, "Other Software Development Functions", discusses any other functions involved in the software development effort not previously addressed. Section 9, "Notes", provides a list of acronyms and glossary. Table 1.3-1, SDP Section and Appendix Outline, depicts the overall outline of the SDP and its appendices.

TABLE 1.3-1 SDP SECTION AND APPENDIX OUTLINE

SECTION	APPENDIX
1.0 Scope	A CSCI List
2.0 Reference Documents	B Software Metrics
3.0 Software Development Management	C Software Engineering Environment Identification
4.0 Software Engineering	
5.0 Formal Qualification Testing	
6.0 Software Product Evaluation	
7.0 Software Configuration Management	
8.0 Other Software Development Functions	
9.0 Notes	

1.4 RELATIONSHIP TO OTHER PLANS

This SDP expands upon and refines the software approaches established in the Program Execution Plan (PEP), D684-10044-1 and Engineering Management Plan (EMP), D684-10014-1. Sections of the SDP relating to Software Configuration Management (SCM), Software Quality Assurance (SQA), Risk Management, Interface Control and Verification and Test include references to the Configuration Management Handbook (CMH), SSP 50123, Prime Safety and Mission Assurance (S&MA) Plan, D684-10700-1, ISS Risk Management Plan, SSP 50175, Prime Interface Control Plan (ICP), D684-10018-1 and Program Master Integration & Verification Plan (PMI&VP), D684-10020-1, respectively, which contain more detailed information pertaining to these functions.

The S&MA plan will contain the Software Quality Program Plan (SQPP) to define SQA's role in the software development process. The SQPP is developed in accordance with the tailored

DOD-STD-2168. This standard applies to the SQA process only and does not dictate the development activities defined under DOD-STD-2167A and this plan.

BSDGs are chartered with developing ISS Program CSCIs. The following development groups are currently allocated responsibility for developing U.S. segments of the ISS software: BSDGs responsible for development of U.S. segments of the ISS software are located at Huntington Beach, CA (B-HB); Canoga Park, CA (B-CP); Huntsville, AL (B-HSV); and Houston, TX (B-HOU). Subsequent references to BSDGs will include one or more of the aforementioned development groups. Rev B

Each BSDG will develop an SDP based on the methods and guidelines identified in this Plan and tailored to the uniqueness of the BSDG's products and development and test environments. All exceptions to this SDP and the SSPS are to be documented in the "Notes" section (Section 9) of the BSDG's SDP. Each developing group is required to abide by the methods contained in their BSDG's SDP. The BSDG's SDPs will be approved by the SWI IPT.

The development group SDP document titles and numbers are as follows:

A.	B-HB, MDC 94H0503	Software Development Plan PG-1;
B.	B-CP, RI/RD 94-601	Product Group 2 Software Development Plan;
C.	B-HSV, D683-10138-1	PG-3 Software Development Plan; and
D.	B-HOU, D684-10085-1	Command and Control Software Development Plan.

This SDP provides overall guidelines and methodologies, but in certain sections does not follow the DID instructions to identify products and items such as non-developmental software. The development group SDPs should follow the format used in this SDP, but provide more detailed information pertaining to the products, environments, methods, etc. to be utilized. Once the development group SDPs are placed under Class I control, any modifications must be pre-approved by the SWI IPT prior to implementation. Figure 1.4-1, Program Documentation Interrelationship, shows the interrelationship between this SDP and other Space Station program plans. In the event of a conflict between this document and higher level documents, the higher level documents take precedence. This same precedence applies at the development group level.

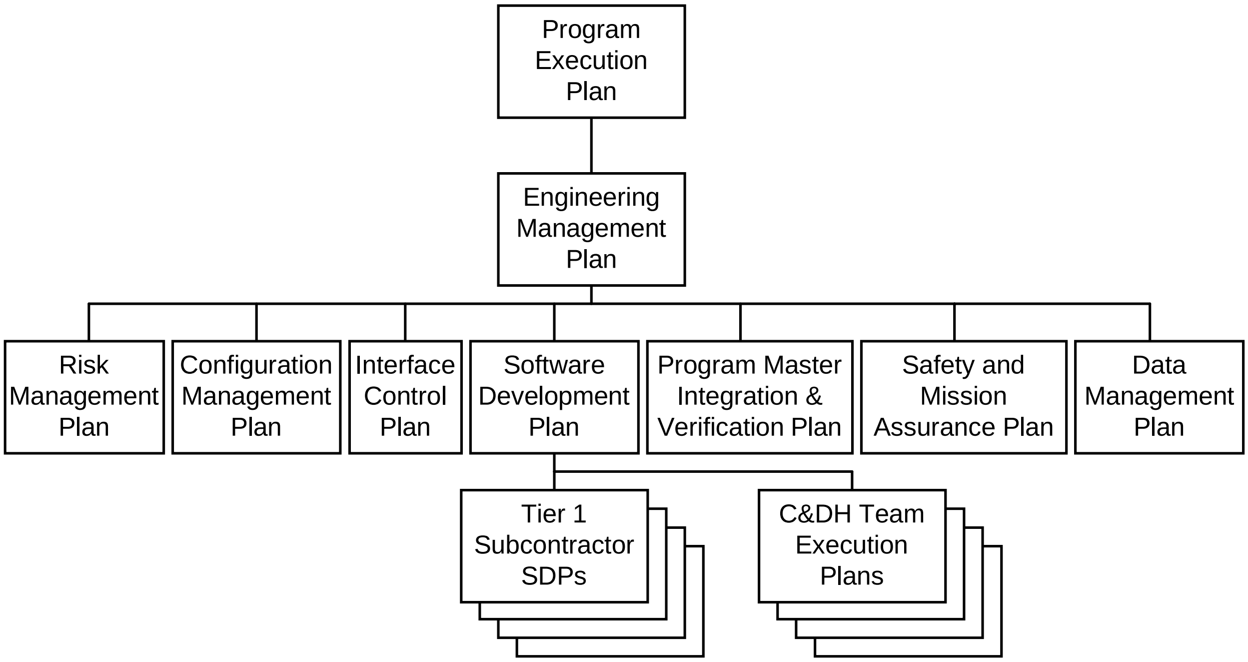


FIGURE 1.4-1 PROGRAM DOCUMENTATION INTERRELATIONSHIP

This page left blank intentionally.

2 REFERENCE DOCUMENTS

This section lists all documents referenced in this SDP.

MILITARY STANDARDS

DOD-STD-2167A	Military Standard, Defense System Software Development. 29 February 1988
DOD-STD-2168	Military Standard, Defense System Software Quality Program. 29 April 1988
MIL-STD-490B	Military Standard, Preparation of Program Unique Specifications. 22 April 1991
MIL-STD-1521B	Technical Reviews and Audits for Systems, Equipment, and Computer Software. 4 June 1985
ANSI/MIL-STD-1815A	Reference Manual for Ada Programming Language. 1983

NASA

JSCM 2410.11	AIS Security Manual. October 1992
SSP 30695	Acceptance Data Package Requirements Specification, Revision A, 26 October 1998
SSP 41153	Prime Software Interface Control Document
SSP 41154	Software Interface Control Document, Part 1, United States On-Orbit Segment to United States Ground Segment, Command and Telemetry. July 1996
SSP 41170	Configuration Management Requirements
SSP 41171	Preparation of Program Unique Specifications. 23 March 1994
SSP 41175-2	Software Interface Control Document, Part 1, Station Management and Control to International Space Station, Book 2, General Software Interface Requirements. May 1996
SSP 50010-01	Documentation Requirements, Standards and Guidelines Vol. 1: Requirements and Standards. July 1994
SSP 50094	NASA/RSA Joint Specifications/Standards for the ISSA Russian Segment
SSP 50123	Configuration Management Handbook
SSP 50134	ISS Risk Summary Card. 10 February 1995

SSP 50175	ISS Risk Management Plan. 5 July 1995

BOEING

D683-10138-1	PG-3 Software Development Plan
D684-10002-1	Space Station Data Management Plan
D684-10014-1	Space Station Engineering Management Plan. 24 March 1994
D684-10018-1	Prime Interface Control Plan
D684-10020-1	Program Master Integration and Verification Plan
D684-10025-1	Prime Integration and Verification Implementation Plan for U.S. Segments
D684-10041-1	Prime Integrated Logistics Support Plan
D684-10044-1	Space Station Program Execution Plan. 13 July 1995
D684-10056-1	Prime Contractor Software Standards and Procedures Specification. Current Revision
D684-10085-1	Command and Control Software Development Plan
D684-10092-1	ISS Prime Contractor Software Interface Control Document SVF Simulation
D684-10097-01	Guidelines and Procedures for the Conduct of Functional Configuration Audit (FCA)/Physical Configuration Audit (PCA) Rev B
D684-10177-1	Mission Build Facility Standard Output Definition
D684-10293-01	Software Configuration Management Handbook
D684-10500	Command and Data Handling Architecture Notebook
D684-10700-1	Prime Safety and Mission Assurance Plan
S684-10003	Space Station System Specification
S684-10140	Prime Item Development Specification for Software Verification Facility. 31 March 1995.
N/A	C&DH Software Integration IPT Team Execution Plan

OTHER

CDRL A002	Interim IV&V Master Plan for ISSA. 15 Sept 1994

MDC 94H0503	Software Development Plan PG-1
RI/RD 94-601	Product Group 2 Software Development Plan
X8264868	MDM User’s Guide

This page left blank intentionally.

3 SOFTWARE DEVELOPMENT MANAGEMENT

The software for the U.S. segments of the ISS will be developed in accordance with DOD-STD-2167A as tailored by this SDP and the Prime SSPS. The Tier 1 Subcontractor reduced documentation set is shown in Figure 3-1, Tailored Software Documentation. The detailed tailoring of the DIDs associated with these documents is contained in the SSPS. This document set and associated data will be required from each Tier 1 Subcontractor for each CSCI being developed with the following exceptions:

A. A single SDP, following DI-MCCR-80030A/T, to cover all the software products being delivered under their contract. At the Tier 1 Subcontractor's discretion, software standards and processes may be included in an SSPS following contractor format and made available with the SDP.

B. A single Software Test Plan (STP), following DI-MCCR-80014A/T, to cover all the formal software tests to be performed under their contract.

C. A Software User's Manual (SUM), following DI-MCCR-80019A, for CSCIs which include user interfaces.

D. A Firmware Support Manual (FSM), following DI-MCCR-80022A, only for those CSCIs residing in Firmware Controllers (FCs). Multiple FC CSCIs developed by the same Tier 1 Subcontractor for the same processor type may be combined into a single FSM.

E. A Software Programmer's Manual (SPM), following DI-MCCR-80021A, only from processor hardware developers.

F. A single quarterly Software Status Report, in contractor format, to include the metrics information specified in Section 3.12 and Appendix B.

G. A single Instrumentation Program and Command List (IP&CL) following the format specified in Supplier Data Sheet (SDS) SS-VE-031, to include all measurement and command data required to support the on board command and control processor and its associated software, on board display and its associated software and Ground Test and Operations facilities.

H. A Software Test Description (STD) Volume 1, following DI-MCCR-80015A/T, to document the formal software test cases and requirements allocation. Test procedures will not be included in this document. This document is not required for flight simulation software CSCIs.

The following documents are required for each CSCI and are briefly described below:

A. A Software Requirements Specification (SRS), following DI-MCCR-80025A/T, for software and interface requirements. This document contains the requirements to be verified in the software Formal Qualification Test (FQT). The SRS is also referred to as a B5 specification as defined in SSP 41171, Preparation of Program Unique Specifications, which supersedes MIL-STD-490.

B. A Software Product Specification (SPS), following DI-MCCR-80029A/T, to document the software design, requirements allocation and as-built software product. The tailored

SPS, VDD and Software Development Folders (SDFs) identified below, will represent the total software design.

C. A Software Test Report (STR), following DI-MCCR-80017A/T, for each formal software test completed. This report will include the "as-run" test procedures.

D. A Version Description Document/Drawing (VDD), following DI-MCCR-80013A, for each software delivery. The source, executable code and data for the CSCI being delivered will be included with the VDD.

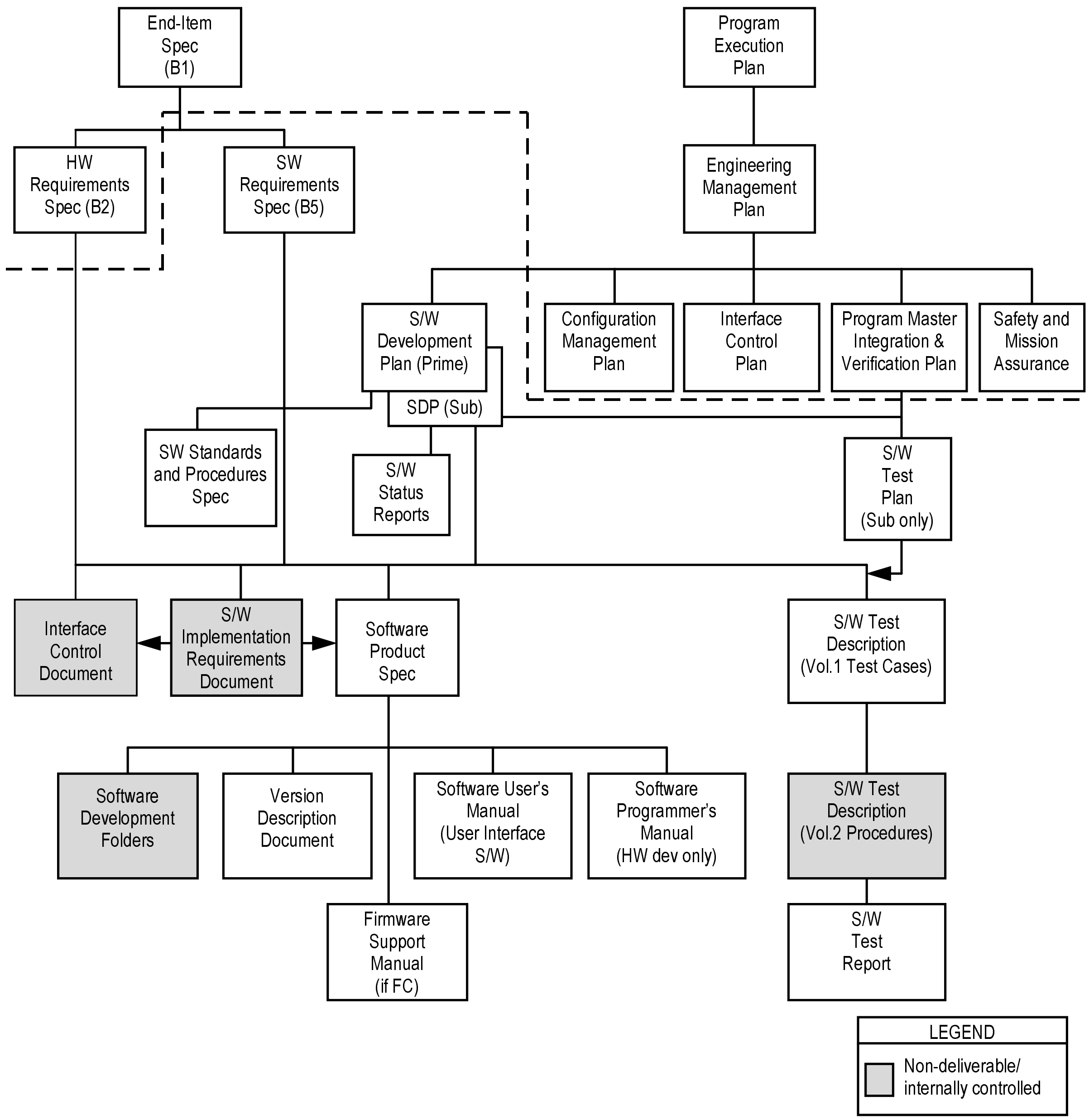


FIGURE 3-1 TAILORED SOFTWARE DOCUMENTATION

The following documents will be maintained by the subcontractors for each CSCI and made available for review by the Prime, NASA, and the Independent Verification and Validation (IV&V) contractor upon request, but are not formal contract deliverables.

A. Interface Control Documents (ICDs) for the user buses, in contractor format to include protocol and transaction sequences with word and bit level data content specified for all external processor input/output.

B. A Software Implementation Requirements Document (SIRD), if developed, will be in contractor format for providing implementation details from the SRS. The SIRD contains non-binding recommendations used for development of the software design and is not part of the binding requirements hierarchy.

C. SDFs to include the detailed design, test, and other information as specified in the SSPS.

D. A STD Volume 2, following Section 10.1.6.1.1.6 of DI-MCCR-80015A, to document the formal software test procedures. The data from this document will be included as part of the final STR.

New software CSCIs/Computer Software Components (CSCs) created or requiring more that 20% redesign following the ISS redesign will be developed and documented in compliance with the DOD-STD-2167A tailoring defined in this plan with the exception of SRSs. Each CSCI will have an SRS. For functionality that has not changed from the SSFP, the ISS SRSs can refer to the SSFP Flight System Software Requirements (FSSRs) to define the software requirements. Code modifications which do not exceed the 20% margin of redesign can be documented and developed using existing standards and guidelines, with the C&DH SWI IPT's approval.

The 20% redesign margin will be determined by a comparison of the existing Source Line of Code (SLOC) count against a SLOC estimation of the effort required to implement the new/changed requirements. C&DH SWI IPT approval of this comparison will be required before the subcontractor can follow a modification to the life-cycle specified in this plan.

The intent of the Prime is to utilize existing documentation and eliminate duplication whenever possible. If software is used in multiple segments, the Tier 1 Subcontractor can choose to provide it in the first segment to be developed and refer to it in a later segment, specifically identifying how this product is used differently. If software is reused within multiple CSCIs, it can be detailed in one CSCI and referred to by other CSCIs. Details of this tailoring will be provided in the Tier 1 Subcontractor SDP. When references are used to satisfy documentation requirements, the appropriate cross reference information must be provided.

In instances where the subcontractor's current software documentation is in a different format than specified above and will require less than 20 percent SLOC update for ISS technical content changes, the existing format is acceptable upon approval, but must be clearly identified in the Tier 1 Subcontractor SDP.

The software deliverables to NASA, as specified in the Prime Statement of Work (SOW), are the SDPs, Prime Software ICDs and Integrated Signal List (ISL), which is a compilation of all the IP&CL data provided by the developers. The VDDs are included as part of the Acceptance Data Package (ADP) (refer to SSP 30695, Rev A, Acceptance Data Package Requirements

Specification). All other software documentation is retained at either the Prime or Subcontractor's facilities, as appropriate, and is available for program review as requested and will be provided to NASA as residual products. Table 3-1, Software Documentation Matrix, depicts the documentation schedule for the Prime and Tier 1 Subcontractor Flight MDM and Simulation Software deliverables.

TABLE 3-1 SOFTWARE DOCUMENTATION MATRIX

Document	DRD	Prep By	Type *	Appl CSCI	Delivery	Developer's Control Level	Developer's Control When	Format
Prime Contractor								
SDP	VE29	D	2	All	SRR/SDR	I	SDR	SSP 50010
SSPS	N/A	D	N/A	All	N/A	N/A	N/A	Contractor
VDD	PC08 ADP	D	3	All	With S/W Delivery	II	ISTRR	Contractor
ICD	VE05/ VE06	B	2	All	SSR/PDR/CDR	I	Part 1 ISSSR Part 2 ISPDR	Contractor
ISL	VE28	D	3	All	With S/W Delivery	II	ISTRR	Contractor
Tier 1 Subcontractors								
SDP	VE33	D	1	All	SDR+75	I	SDR+75	SSP 50010
Requirements								
SRS	SW001	C	2	All	SSR	II	NLT CDR-30	Contractor
SIRD	NR	C/ND	N/A	N/A	Residual	N/A	N/A	Contractor
Design								
ICD (User Bus)	NR	C/ND	N/A	All	Residual	N/A	N/A	Contractor
FSM	SW010	C	3	All FCs	FCA/PCA	II	PCA	Contractor
SPS	SW006	C	2	All	Initial PDR Final FCA/PCA	II	PCA	Contractor
SDF	NR	C/ND	N/A	All	Residual	N/A	N/A	Contractor
SUM	SW008	U	3	User I/F	FCA/PCA	II	PCA	Contractor
VDD	VE35	C	2	All	With S/W Delivery	II	PCA	Contractor
IP&CL	VE31	D	2	All	Quarterly SDR+75	II	PDR	Contractor
SPM	SW007	P	3	MDM	FCA/PCA	II	PCA	Contractor

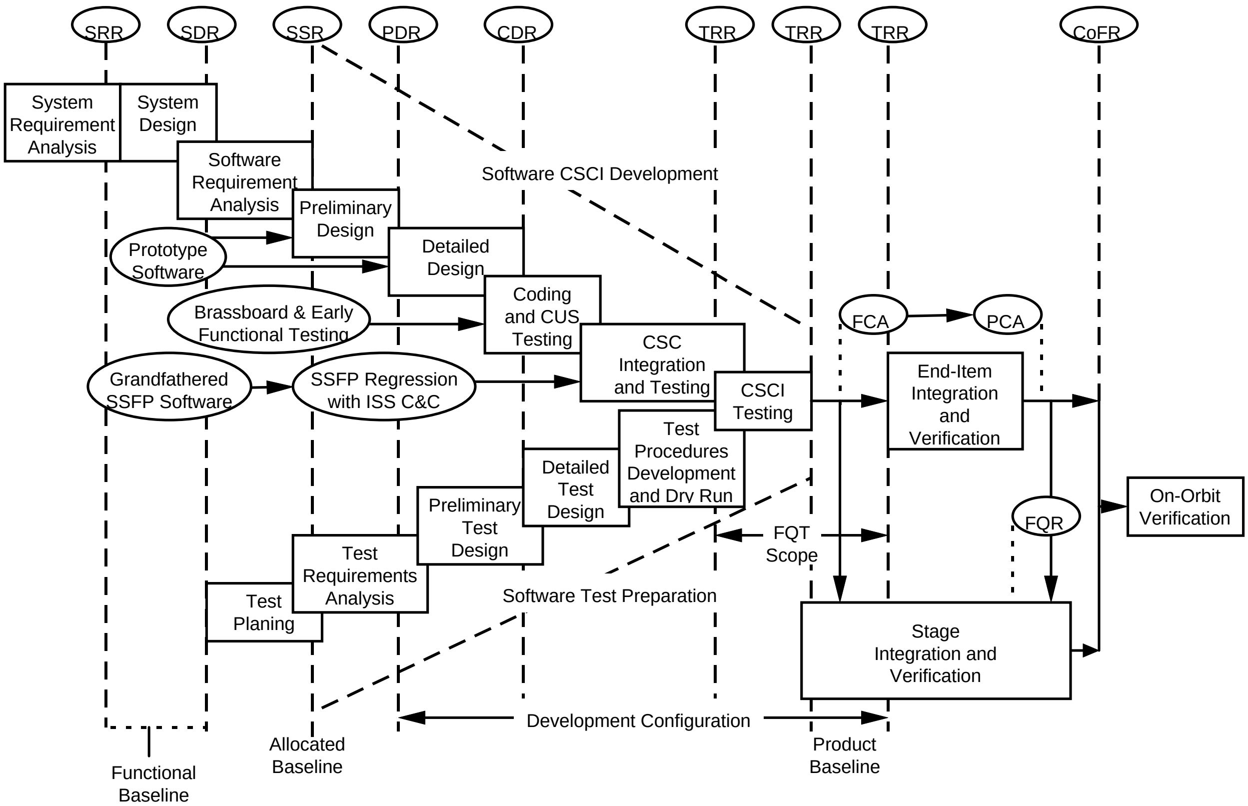
TABLE 3-1 SOFTWARE DOCUMENTATION MATRIX (Concluded)

Document	DRD	Prep By	Type *	Appl CSCI	Delivery	Developer'sControl Level	When	Format
Testing								
STP	SW002	D	2	All	PDR	II	1st CDR-30	SSP 50010
STD (Vol. 1 Test Cases)	SW003	C	2	All except flight sims	CDR	II	TRR-30	Contractor
STD (Vol. 2 Procedures)	NR	C/ND	N/A	All	Residual	N/A	N/A	Contractor
STR	SW004	C	2	All	Test Comp +30	N/A	N/A	Contractor
Reports								
Software Status Reports	SW005	D	3	All	Quarterly, SSR +45	N/A	N/A	Contractor
D - 1 only prepared by each Contractor (Developer)					C - 1 prepared for each CSCI			
P - 1 only prepared by each Processor Contractor					ND - Non-deliverable/Internally controlled			
U - 1 only prepared for each User Interface					NR - Not Required			
I - Formal Customer Control					B - Developed by the Prime for the top 2 tiers of MDMs			
II - Formal Developer Control					NLT - No Later Than			
* Type for the Prime Contractor documents refers to the data type of deliveries to NASA as defined in SSPS 50010-01, Documentation Requirements, Standards and Guidelines. Type for the Tier 1 Subcontractor documents refers to the data type of deliveries to the Prime.								

Ground Support Equipment and Firmware Controller documentation will be delivered to the Prime one time at Functional Configuration Audit/Physical Configuration Audit (FCA/PCA). As this documentation is being developed, it will be available for review at the Developer's facility. Initial and final deliveries of all documentation are electronic. Updates to the documentation can be maintained as redlines until released electronically as block updates as determined by the Developer.

The ISS management approach is a product-oriented team structure which provides clear lines of authority and accountability. Each product is managed by a team of government, contractor and subcontractor personnel, responsible for its success. The Space Station Analysis Integration Team (AIT)/IPT process establishes the architecture and integration approach for development of the Space Station. The Space Station AIT provides technical coordination for inter-IPT concerns. Each IPT is responsible for a specific end item deliverable as documented by a Prime Item Development Specification (PIDS).

The software development approach defined in more detail in Section 4.2 of this plan, combines rapid-prototyping of critical or high-risk functions with the classical waterfall software life-cycle defined in DOD-STD-2167A. Figure 3-2, Tailored Software and Formal Qualification Testing Life-Cycle, depicts this combined life-cycle. Each Tier 1 Subcontractor will provide the detailed implementation of this approach in their SDP.



REVIEWS

SRR - System Requirements Review
SDR - System Design Review
SSR - Software Specification Review
PDR - Preliminary Design Review
CDR - Critical Design Review

TRR - Test Readiness Review
FCA - Functional Configuration Audit
PCA - Physical Configuration Audit
FQR - Formal Qualification Review
CoFR - Certificate of Flight Readiness

FIGURE 3-2 TAILORED SOFTWARE AND FORMAL QUALIFICATION TESTING LIFE-CYCLE

The standards and procedures to be used by the Tier 1 Subcontractors in performing their software and data development are defined in the SSPS. The SSPS also provides the document standards, including the tailored DOD-STD-2167A DIDs and contents for the SDFs. Each Tier 1 Subcontractor will document any variance to the Prime SSPS in Section 9 of their SDP.

Due to the diversity of products developed and the current state of development, it is anticipated that each type of software developed under the guidelines of this SDP will follow a variation of this life-cycle and produce a subset of the products identified in Table 3-1. Figure 3-3, Program Phasing Relationships, illustrates the deliverables from the Prime to NASA and from the Tier 1 Subcontractors to the Prime at each specified program milestone.

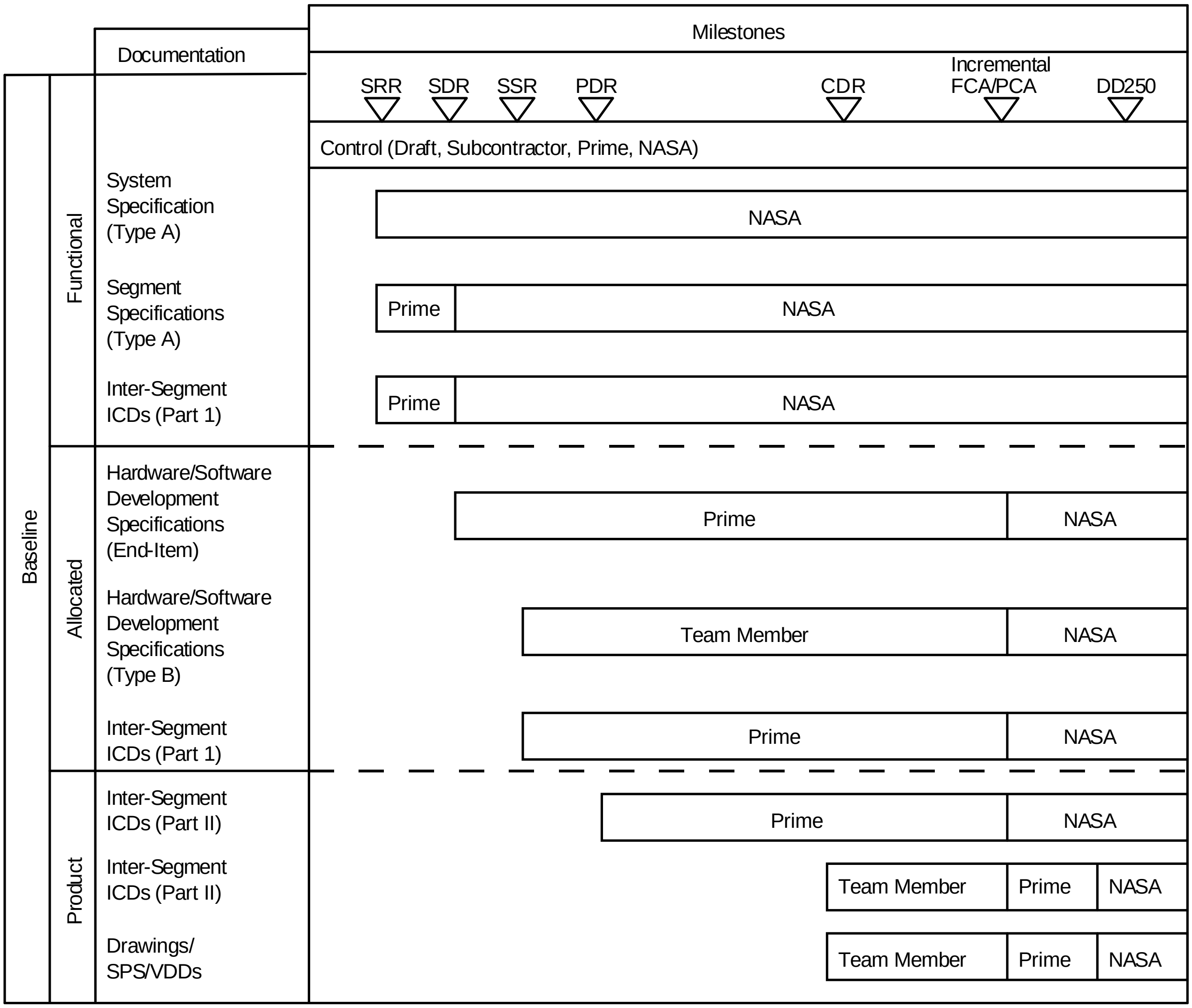


FIGURE 3-3 PROGRAM PHASING RELATIONSHIPS

3.1 PROJECT ORGANIZATION AND RESOURCES

The ISS Program is a partnership among NASA, the IPs, and private industry. The NASA Program Manager directs the Prime and interfaces with the program offices of the IPs and other agencies external to the Space Station program. The Prime is responsible for the development and delivery of the U.S. On-orbit end items of the Space Station, as well as technical coordination with the IPs and external NASA programs.

3.1.1 CONTRACTOR FACILITIES

The Space Station is being developed by companies widely dispersed geographically. Therefore, the facilities to develop software are also geographically separated. To support the Space Station IPT approach in the most effective manner possible, the Prime is collocated with the NASA Program Office in Houston, TX to manage the development of the Space Station software. However, the development and integration of the individual end item software requires varying facilities at the Tier 1 and Tier 2 Subcontractor's sites. These facilities vary in capability based on the need of the individual software being developed.

The Tier 1 Subcontractors will have Tier 2 Subcontractors at varying locations which will be identified in their Tier 1 SDP. The SVF is the site of the horizontal software integration and test and is located in NASA's Sonny Carter Training Facility in Houston, TX.

3.1.2 GOVERNMENT FURNISHED EQUIPMENT, SOFTWARE, AND SERVICES

The processes, standards, procedures and practices identified in this SDP do not apply to GFE software unless changes are required to this software or the NASA Program Office directs otherwise. Should NASA direct the Prime to modify GFE software, the extent to which this SDP applies will be determined on a case by case basis. At the present time, GFE software identified for the program includes:

A. Payload General Support Computer Software for handling payload commanding telemetry;

B. Timeliner User Interface Language (UIL) onboard kernel and onboard adapter;

C. UIL compiler kernel and adapter;

D. Habitation medical equipment firmware controllers;

E. Ultrahigh Frequency (UHF) Communications Subsystem Transmitter;

F. Portable Computer System;

G. Orbiter Interface Unit system;

H. Global Positioning System (GPS) Receiver/Processor flight firmware;

I. Columbus Ground Software tools; and

J. International Partner computer systems.

Any additional GFE software used by the various Tier 1 Subcontractors will be identified in their SDPs. Any GFE products will be handled and distributed to the Tier 1 Subcontractors by the Prime as depicted by Figure 3.1.2-1, GFE/CFE Product Process Flow. The schedule for delivery of GFE items is identified in the Prime GFE List (GFEL). Software and equipment furnished by a Tier 1 Subcontractor for use by another Tier 1 Subcontractor will be classified as Contractor Furnished Equipment (CFE) and identified in the CFE List (CFEL) included in each of the Tier 1 Subcontractor's contracts. Delivery of the actual CFE products will be made directly between the providing and receiving Tier 1 Subcontractors. The Tier 1 Subcontractor delivering the product will provide documentation confirming these deliveries (e.g., DD Form 1149 or commercial shipper) to the Prime to ensure contract compliance.

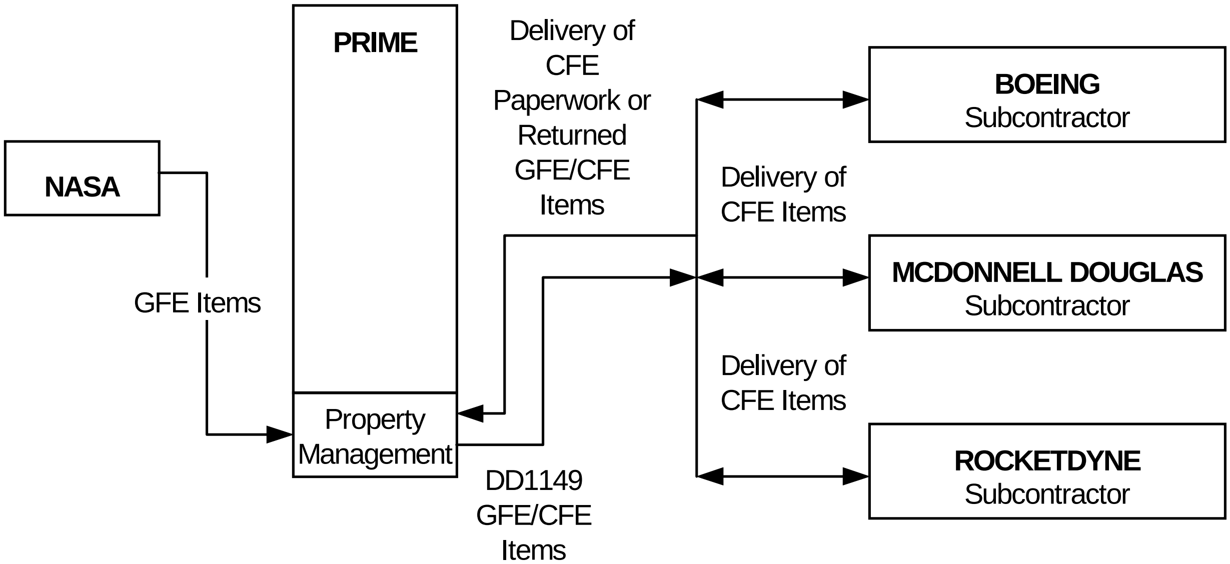


FIGURE 3.1.2-1 GFE/CGE PRODUCT PROCESS FLOW

3.1.3 ORGANIZATIONAL STRUCTURE

Figure 3.1.3-1, C&DH Subsystem IPT Organization, shows the structure of the C&DH Subsystem Provider IPT. The C&DH Subsystem Provider IPT is responsible for:

A. Development and verification of the C&DH subsystem consistent with the ISS system, segment and end item specifications, C&DH architecture and launch sequence;

B. Providing C&DH software and hardware components to Launch Package IPTs for integration into U.S. flight elements and to ground facilities to support development, integration, verification, and training;

C. Providing hardware and software to IPs as required; and

D. Developing and verifying ground hardware and software that supports development and maintenance of C&DH software and related databases.

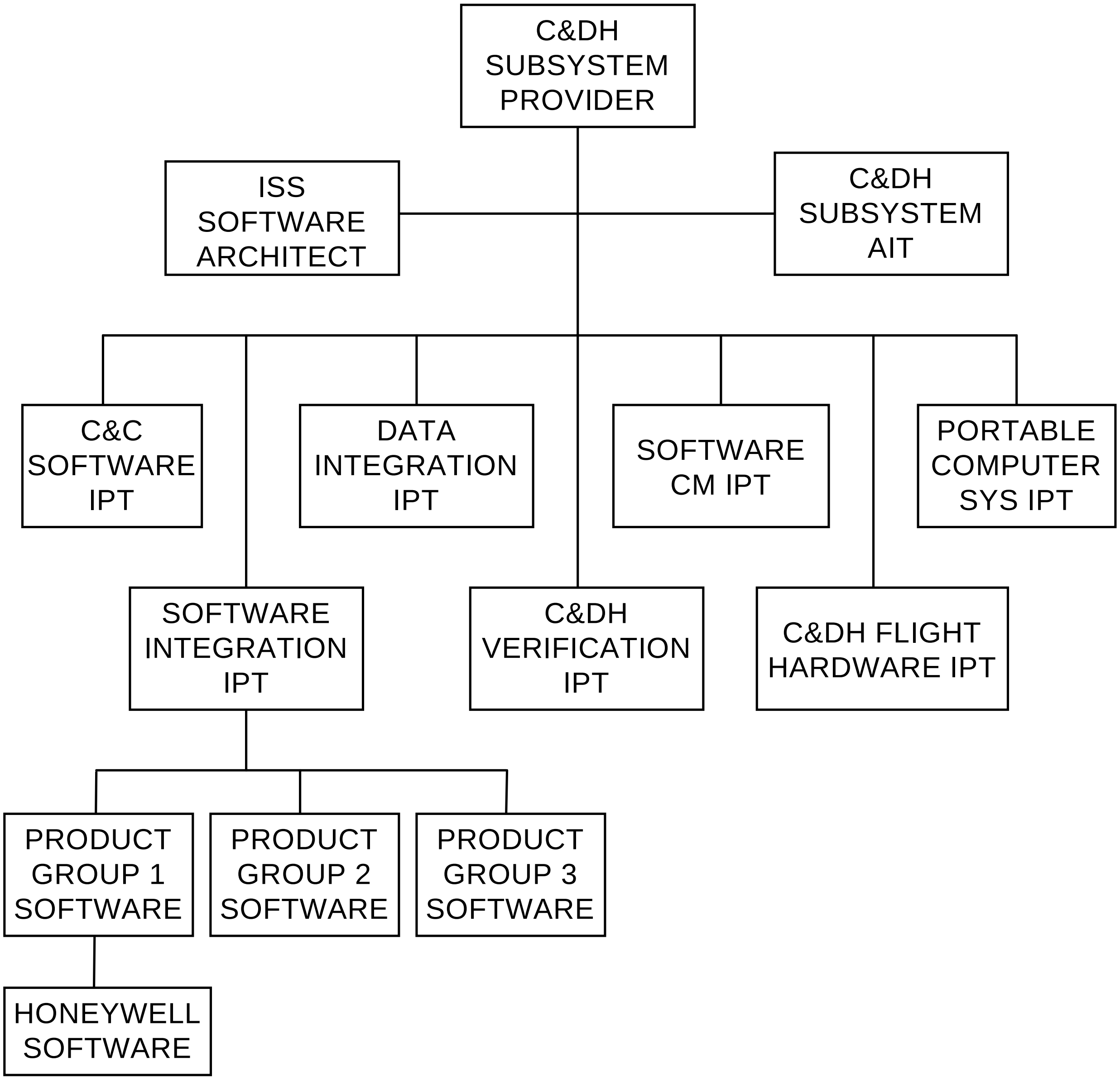


FIGURE 3.1.3-1 C&DH SUBSYSTEM IPT ORGANIZATION

The C&DH Subsystem Provider IPT is a part of the Vehicle Product Team and reports to the Subsystems Provider IPT Team Leader. Flight and developmental hardware and software produced by the C&DH Subsystem Provider IPT are delivered to the Launch Package IPTs and various integration test and training facilities. The C&DH Subsystem Provider IPT also provides requirements definition, analysis and test conduct support to the Vehicle AIT (VAIT).

The C&DH Subsystem Provider IPT is led jointly by a Boeing and a NASA manager and includes the C&DH Subsystem Provider sub-team leaders and a C&DH manager from each Tier 1 Subcontractor. In addition to regular IPT meetings, this team will meet regularly in Houston for a C&DH Management Team meeting to discuss status and C&DH unique issues.

Coordination among the Prime, NASA and the Tier 1 Subcontractors is focused on the C&DH subsystem level within the C&DH Subsystem Provider IPT. Interaction among these parties will also take place at the C&DH sub-team level, focusing on the corresponding C&DH component and/or the integration of components (e.g., CSCI integration for specific ISS stages is coordinated by the SWI IPT).

Like the C&DH Subsystem Provider IPT, each of the sub-teams is led jointly by a Boeing and a NASA leader. The sub-teams include members representing the Tier 1 Subcontractors as appropriate.

3.1.3.1 C&DH SUBSYSTEM AIT

The C&DH Subsystem AIT has the responsibility for analysis and integration of the end-to-end C&DH product as defined in the Prime contract. The C&DH Subsystem AIT reports to the C&DH Subsystem Provider IPT leaders, but has a direct relationship with the VAIT and Space Station AIT (SSAIT). The C&DH Subsystem AIT is responsible for:

A. Development of the end-to-end C&DH architecture;

B. Development and verification of the end-to-end C&DH design;

C. Development of specification and ICD inputs required from C&DH;

D. Providing C&DH Subsystem performance analysis in support of Vehicle-level analysis;

E. Maintaining the consolidated C&DH risk list; and

F. Serving as the C&DH focal point for technical coordination with IPs.

3.1.3.2 C&DH FLIGHT HARDWARE IPT

The C&DH Flight Hardware IPT has the responsibility for designing, developing, producing, qualifying and providing all C&DH flight and ground support hardware and associated firmware. This IPT interfaces with B-HB and Honeywell Incorporated (HI) for development of the C&DH unique hardware, including the MDM Application Test Environment (MATE) and MDM Functionally Equivalent Units (FEUs). Rev B

3.1.3.3 C&DH DATA INTEGRATION IPT

The C&DH Data Integration IPT has the responsibility for performing analysis and integration tasks in support of C&DH ISS Flight and supporting Ground data development. The tasks associated with this include:

A. Development and coordination of the data standards, including maintenance of the SSPS;

B. Coordination, development and maintenance of the Standard Out Definition document;

C. Coordination with the MBF to define data handling and processing requirements; and

D. Collection and maintenance of the IP&CL data and generation of the ISL.

3.1.3.4 COMMAND AND CONTROL SOFTWARE IPT

The Command and Control (C&C) Software IPT is responsible for the design, development, qualification, delivery and support of the Prime developed Command and Control Software (CCS) and Node 1 Control Software (NCS) flight and simulation software and associated documentation. They are also responsible for supporting the Station Management and Control (SM&C) IPT in requirements definition and development for the CCS and NCS flight software. The C&C Software IPT provides its software products to the SWI IPT and Data Integration IPT as if it were a Tier 1 Subcontractor.

3.1.3.5 C&DH SOFTWARE INTEGRATION IPT

The C&DH SWI IPT provides the development planning, technical oversight, analysis and integration tasks required to insure the integrity of the flight software, and associated documentation and simulations. The SWI IPT is a Prime/NASA led team in Houston; the Product Group (PG) teams are located at the respective Tier 1 Subcontractor location and led by that subcontractor. The C&DH SWI IPT is responsible for:

A. Development and maintenance of this SDP;

B. Coordination of software development and formal test activities to ensure compliance with the SDP requirements and qualified delivered products;

C. Integrated plan and schedule development/maintenance ensuring software deliveries in support of launch manifest;

D. Development of the C&DH software architecture and control concept; and

E. Integration of all C&DH flight software products.

Although the C&C Software IPT organizationally is not within the SWI IPT, this IPT has the same coordination responsibility for the CCS/NCS software as it does for PG software.

3.1.3.6 PORTABLE COMPUTER SYSTEM IPT

The NASA PCS Project Office IPT has the responsibility for defining and developing the crew interface system. This includes selection of the PCS, development of the PCS software and development of the displays crews will use to interface with the system. Rev B

3.1.3.7 SOFTWARE CONFIGURATION MANAGEMENT IPT

The SCM IPT has the responsibility for identification and application of the SCM disciplines for maintaining configuration control of all C&DH software and data delivered to the Prime. This IPT also provides in-process SCM support to the C&C Software IPT.

3.1.3.8 SOFTWARE VERIFICATION IPT

The Software Verification IPT has the responsibility for the design, development and implementation of a C&DH subsystem verification program. This includes test planning, test documentation preparation and test conduct for each Stage. The C&DH verification activity is part of the vehicle verification activity coordinated under the Integrated Test and Verification AIT of the VAIT.

3.1.3.9 ISS SOFTWARE ARCHITECT

The Software Architect is the primary expert developing the concepts and framework for the C&DH software architecture, control concepts and data transfer. The Software Architect works closely with the SWI IPT members who expand his concepts to develop detailed software protocols and requirements. The Software Architect's tasks include:

A. Providing software architecture technical expertise to the teams developing ISS software;

B. Principal integrator for all control concepts:

C. Principal technical interface for the Control Center developers and users; and

D. Provides technical support in the development of IP software.

3.1.4 PERSONNEL

Each IPT leader whose products include software will have on his team all the disciplines necessary to completely develop the software product and all supporting documentation and testing systems. A list of the personnel, including the discipline they are representing, is kept current in each sub-team's Team Execution Plan (TEP). The C&DH Subsystem Provide IPT sub-team TEPs are updated at least yearly and approved by the C&DH Subsystem Provider IPT team leaders.

3.2 SCHEDULE AND MILESTONES

The Prime is responsible for establishing and implementing the methodologies for effective integration of the Tier 1 Subcontractor's schedules. The objective of the schedule integration process is to review and analyze the Tier 1 Subcontractor schedules, as appropriate, to achieve overall integration of program plans. The integrated program plans are reflected in the Prime software schedules.

The detailed software schedule information, implementation plans, and internal subcontractor reviews are documented and maintained by the Tier 1 Subcontractors in their respective forms (e.g.; PC06) and are not included in this plan. Program schedule data including launch and contractual reviews, are documented in the PEP and maintained by the Prime. C&DH stage software schedules (that include dependencies between Tier 1 Subcontractors) are developed and maintained by the C&DH SWI IPT. The results of the development and on-going coordination of the dependencies of these schedules are reflected in the delivery dates contained in the Program CFEL, Deliverable Items List (DIL) and GFEL.

The generic integrated stage template is illustrated in Figure 3.2-1, Tailored Software and Formal Qualification Testing Life-Cycle. All issues (including disconnects) are worked and resolved in the C&DH SWI IPT. The C&DH SWI IPT will report software schedule status to the C&DH

Subsystem Provider IPT, for integration into the overall C&DH subsystem integrated schedule. Figure 3.2-2, Schedule Integration Process, illustrates the schedule integration process.

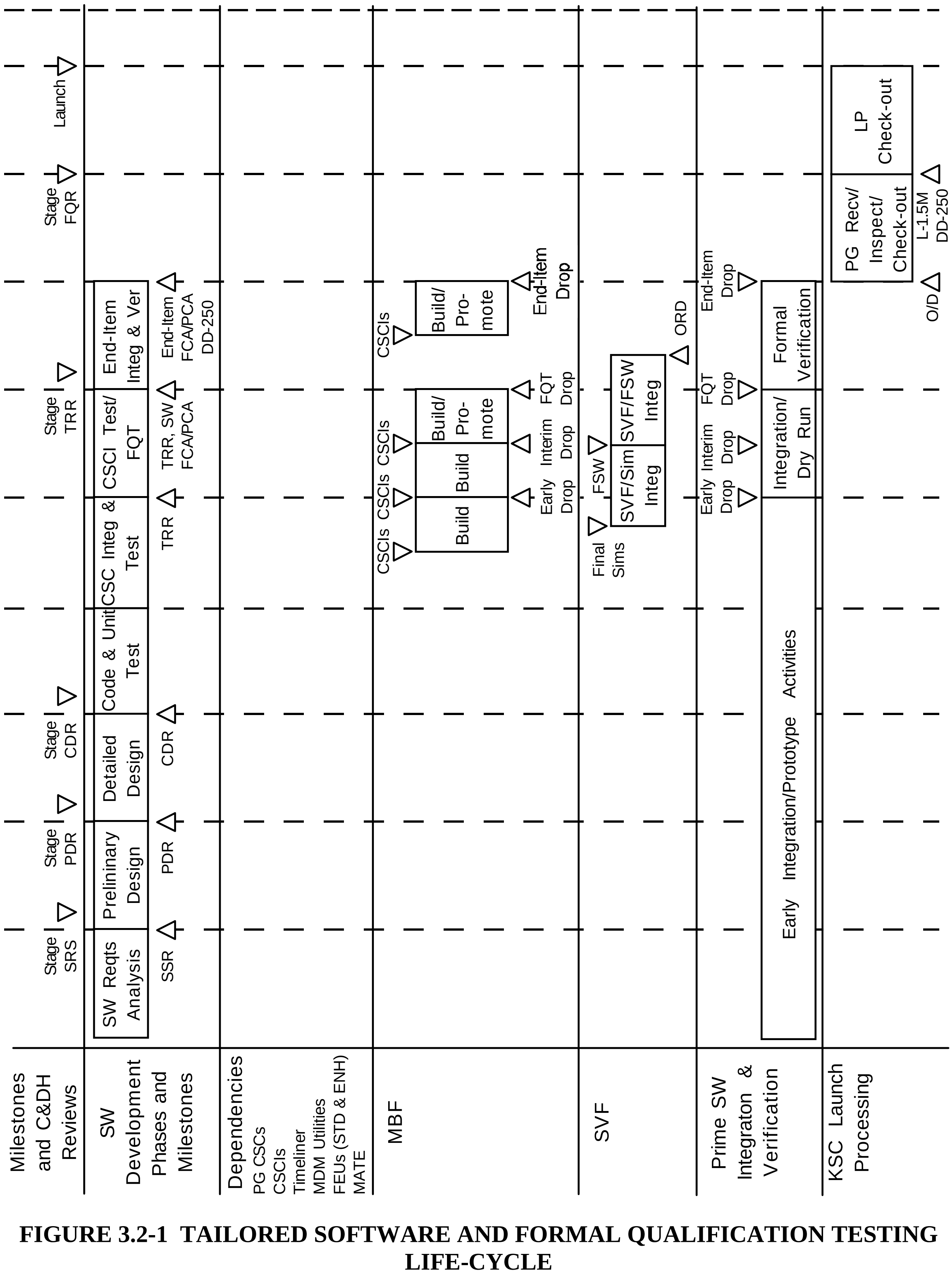


FIGURE 3.2-1 TAILORED SOFTWARE AND FORMAL QUALIFICATION TESTING LIFE-CYCLE

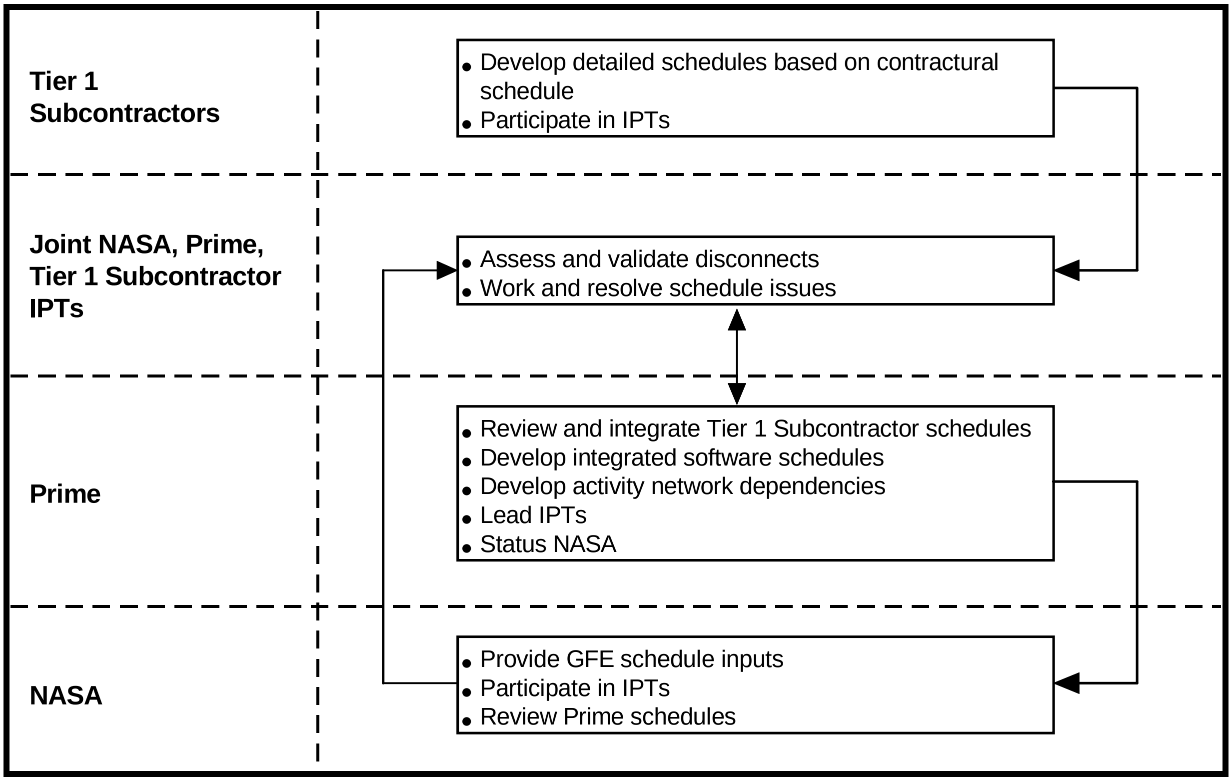


FIGURE 3.2-2 SCHEDULE INTEGRATION PROCESS

3.2.1 ACTIVITIES

The schedule information to be included in the Prime software schedules includes launch dates, program reviews, deliverables as documented in the Subcontractor Data Requirements Lists (SDRLs), and all dependencies between Tier 1 Subcontractors. Detailed information is contained in the Tier 1 Subcontractor schedules.

3.2.2 ACTIVITY NETWORK

Relationships among software development activities and other program activities are identified and networked together at dependent or critical milestones. This information is developed by the C&DH SWI IPT by integrating the activity network software dependencies identified for each Stage. Interdependencies between the Tier 1 Subcontractors are identified and resolved by the C&DH SWI IPT. Each IPT will status changes to the schedules and dependencies to the C&DH SWI IPT as the changes occur.

3.2.3 SOURCE IDENTIFICATION

Each Tier 1 Subcontractor is responsible for providing all deliverables as specified in their SOW, DIL, CFEL and SDRLs, including releasing contract deliverable CSCIs, associated data and documentation. In addition, each Tier 1 Subcontractor is responsible for providing the resources required to support integration activities as specified in this document. All required resources will be coordinated through the IPTs.

3.3 RISK MANAGEMENT

A common risk management approach will be used on the ISS program and is defined in the Risk Management Plan. A summary of the program risk management process, instructions for determining the likelihood and consequence of the risk and method for reporting the risks are included on the ISS Risk Summary Card, SSP 50134. Using this approach and the IPT structure, risk management will be handled at a local level wherever possible. In some cases, an issue will be beyond the ability of an IPT to manage, due to either a lack of resources or a wide-ranging impact. These items should be elevated to a higher-level IPT.

Risk management processes that are unique to each Tier 1 Subcontractor are documented in their individual SDPs. Each risk area will include a description of the risk, probability considerations, consequence considerations (performance, cost, and schedule) and assumptions. The following paragraphs identify the software specific implementation of the risk management approach being used across the program. Figure 3.3-1, Risk Management Process Flow, illustrates the flow of C&DH specific identified risks through the program.

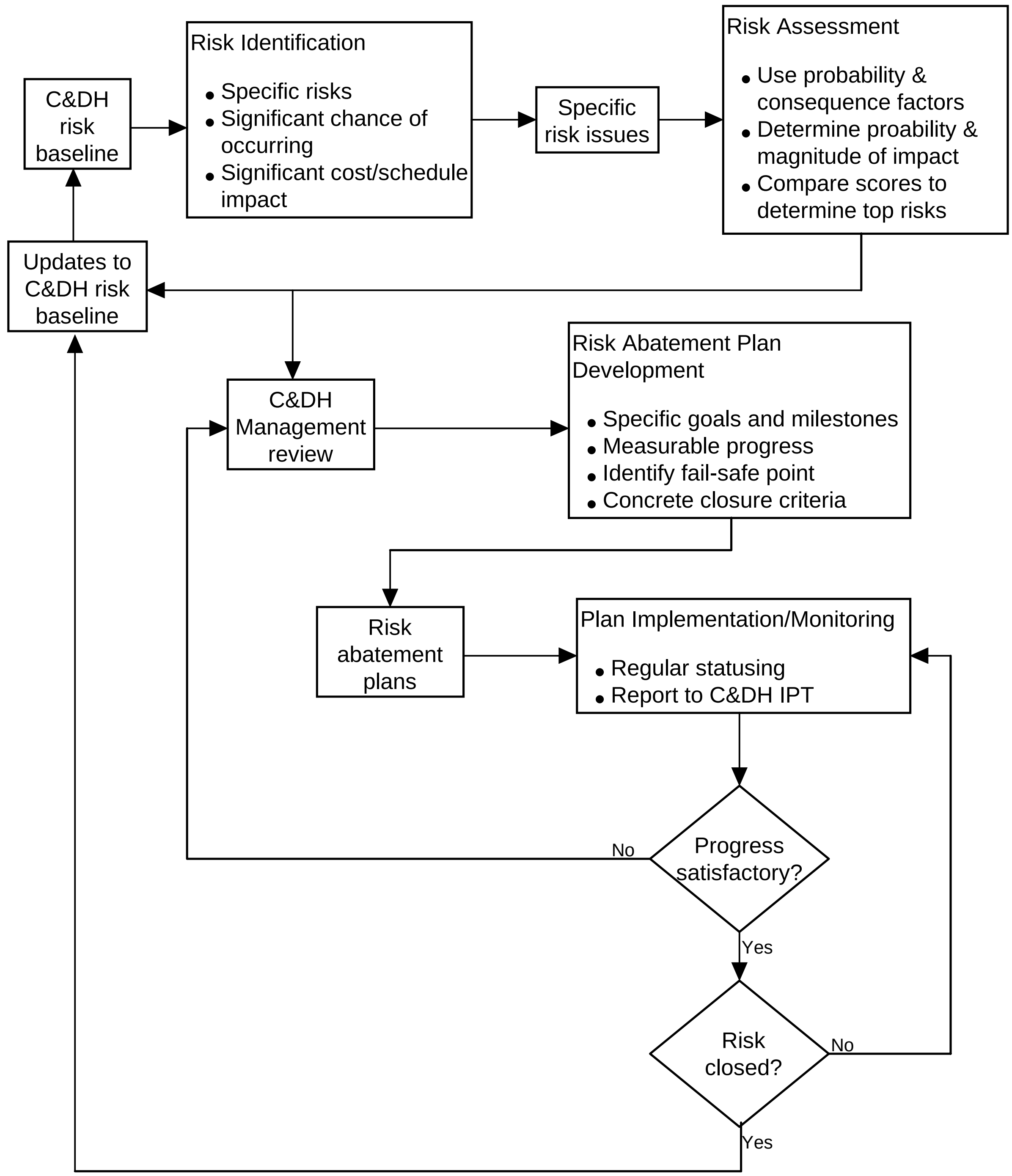


FIGURE 3.3-1 RISK MANAGEMENT PROCESS FLOW

The C&DH Subsystem AIT will maintain a list of consolidated C&DH risks, including software risk items and a risk abatement plan for each item on the list. Software risks will be reviewed at the originating IPT level and if accepted by the C&DH SWI IPT, the risks will be entered into the Risk Data Management Application (RDMA). RDMA provides the capability to elevate the risks to higher levels as required. The C&DH SWI IPT will track and present program critical software risks to the C&DH Subsystem AIT.

3.3.1 RISK IDENTIFICATION

Risk identification is the process of selecting the specific potential risks to assess. Risks may relate to technical, safety, security, resource, schedule, cost, or other areas of concern. Each IPT developing software must develop an original baseline assessment of risk by applying the following scale to the different systems and subsystems under its jurisdiction. The baseline assessment should be reevaluated on a quarterly basis to ensure the results of the assessment are still valid. As ongoing issues are raised, the same scale will be applied and the results compared against the baseline scores. Each of the risk categories use a common set of values to determine the probability of occurrence and the program impacts (in cost, schedule, and performance).

In addition to the ISS Risk Summary Card, the following are guidelines for identifying risks to be assessed:

A. List only specific risks. Do not include generic risks such as, ”the software fails to meet performance requirements”.

B. List only risks which have a significant (i.e., greater than 5%) probability of occurrence.

C. List only risks that would have a significant negative impact if they were to be realized. Assume ”significant” to be greater than a one man-month equivalent cost or schedule impact.

The following subsections identify risk categories and are intended to assist in the determination and review of potential risks. These lists are generic in nature and do not identify all possible risks or classes or risks. There is some overlap between the lists since certain risks have aspects that fall into different categories.

3.3.1.1 TECHNICAL RISKS

Technical risks and potential causes to be considered when developing an initial risk list are the following:

A. Development of incorrect software functions;

B. Development of unneeded software functions;

C. Real-time performance short-falls;

D. Risks due to volatility of software requirements;

E. Development of unclear or incorrect user interfaces;

F. Software that exceeds allocated resources (memory size, timing, etc.);

G. Risks associated with not developing a working prototype;

H. Risks associated with projecting performance based on a prototype;

I. Risks associated with the reuse of software (hidden features, extent of modification required, etc.);

J. Risks associated with inexperience with Ada (tasking problems, etc.);

K. Risks associated with inexperience with Software Engineering Environment (SEE), configuration management, and other Computer Aided Software Engineering (CASE) tools used in development, test, and control of software;

L. Risks associated with inexperience with the design or test methodology;

M. Risks associated with a lack of software maintenance features; and

N. Risks associated with a lack of software diagnostic features.

3.3.1.2 SAFETY RISKS

Safety risks and potential causes to be considered when developing an initial risk list are the following:

A. Risks associated with the inadvertent exceeding of computer resource capabilities;

B. Failure of backup capabilities;

C. Failure of recovery capabilities;

D. Risks associated with fault intolerant software;

E. Risks associated with incomplete or poorly planned software/hardware upgrades; and

F. Risks associated with poorly defined safety requirements.

3.3.1.3 SECURITY RISKS

Security risks and potential causes to be considered when developing an initial risk list are the following:

A. Risks associated with the inadequate control of access to the flight software; and

B. Risks associated with deliberate damage to the software development library.

3.3.1.4 PERSONNEL RESOURCE RISKS

Personnel resource risks and potential causes to be considered when developing an initial risk list are the following:

A. Personnel shortfalls;

B. Incorrect personnel mix;

C. Lack of experienced personnel; and

D. Risks associated with retaining key personnel for an extended development and test period.

3.3.1.5 SCHEDULE RISKS

Schedule risks and potential causes to be considered when developing an initial risk list are the following:

A. Failure to achieve major milestones;

B. Risks due to the number of tasks on the critical path;

C. Risks due to the complexity or severity of project dependencies; and

D. Risks due to premature milestones.

3.3.1.6 COST RISKS

Costs risks and potential causes to be considered when developing an initial risk list are the following:

A. Risks due to an unrealistic budget;

B. Risks due to an uncertain budget;

C. Risks due to uncertainty in the cost estimation method; and

D. Risks due to overspending in order to meet early milestones.

3.3.1.7 COMPUTER RESOURCE RISKS

Development host/target computer risks and potential causes to be considered when developing an initial risk list are the following:

A. Inadequacy of computer resource capabilities;

B. System performance shortfalls;

C. Risks due to a lack of familiarity with the system hardware/software;

D. Risks due to unavailability of system hardware/software; and

E. Risks due to a lack of normal or disaster backup.

3.3.1.8 EXTERNALLY DRIVEN RISKS

Externally driven risks and potential causes to be considered when developing an initial risk list are the following:

A. Difficulties encountered in interfacing with external organizations;

B. Unavailability or inadequacy of provided software/hardware;

C. Risks associated with externally selected software tools;

D. Uncertainties concerning overall project responsibilities and authority; and

E. Vague interface requirements.

3.3.1.9 PROJECT UNIQUE RISKS

Project unique risks and potential causes to be considered when developing an initial risk list are the following:

A. Risks associated with integration of application software from several contractors;

B. Risks associated with the use of custom/customized operating systems;

C. Risks associated with immaturity of operating systems and services; and

D. Risks associated with the use of shared processors.

3.3.2 RISK ASSESSMENT

Each identified risk will be assessed to determine its probability and magnitude of impact on the development and/or operation of the software. To determine the impact to the project, a probability of occurrence value will be assigned to each risk item. The ISSA Risk Summary Card defines how the Prime will assign probability of risk occurrence. Tools and other information useful in risk analysis include historical metrics, software development metrics, analyses performed by the development, test, SQA, Safety, and IV&V organizations, and lessons learned from the SSFP.

3.3.3 RISK ABATEMENT

Once a particular item has been slated for risk abatement efforts, an abatement plan must be developed. Before the IPT can pursue a plan, the possible alternatives must be analyzed and the best method (or combination of methods) must be chosen. Some possible abatement activities are defined in the Risk Management Plan.

Each individual risk item will lend itself to specific abatement activities not mentioned here. It is the job of the appropriate level IPT (Prime, Tier 1 Subcontractor, etc.) to generate as many options as possible. Once the methods have been established, a specific plan of action must be chosen by either management or the whole IPT.

The chosen risk abatement plan must have the following characteristics:

A. It must be specific: Each plan must define a process with a specific set of goals and milestones.

B. Progress must be measurable: The IPT should be able to keep a running status of the abatement plan to evaluate its effectiveness and decide if further action is necessary.

C. It should contain a ”fail-safe” point: There should be a point in the plan where, if progress is unsatisfactory or the risk is unchanged, a ”fallback” method of recovery plan should be enacted.

D. It must have an end point: When the plan is developed, it must contain concrete closure criteria. Some abatement plans will have definite endpoints - they will be finished when a piece of hardware is qualified, when drawings are back on schedule, etc. Some will only be finished when the program is over. Once an IPT considers the risk to be abated, the plan can be closed and the item dropped from the list of active risk items.

Monthly updates are required to status development of abatement plans or to status progress of abatement activities. Updates are to be made by the end of the first week of each month.

3.3.4 ONGOING RISK MANAGEMENT

Once the initial risk abatement plans have been in effect for a period of time (generally a quarter), the IPT should begin another risk management process by using the risk identification method contained herein and constructing a new ranked list of risk items. If the abatement plans have been successful, some of the previous items should have dropped in rank. If any items with abatement plans have risen (i.e., increased in risk), the IPT should consider elevating them to a higher IPT.

3.4 SECURITY

Prime Contractor computer facilities which are used to develop, test, manage, and store ISS flight software, flight data and data used for ground facility reconfiguration will implement access controls, backup procedures, and management processes which protect the software and data against corruption, loss or unauthorized disclosure according to the sensitivity and/or criticality of the software, data and the applications which process them. JSCM 2410.11, the JSC Automated Information Systems (AIS) Security Manual, is used by the Prime as a guide for determining the proper controls and administrative processes for protecting the software and data from the following specific types of threats:

A. Utility disruptions;

B. Natural disasters;

C. Unauthorized access by outsiders; and

D. Unauthorized access by insiders.

For reference, JSCM 2410.11 defines controls based on determination of a sensitivity/criticality level of between 0 and 3 as defined in Table 3.4-1, Sensitivity Levels. Level 0 is essentially uncontrolled. Level 1 is appropriate for a research environment or an E-Mail system. Level 2 covers sensitive information falling under restrictions of the Privacy Act of 1974, or systems whose loss could pose a significant impact on program success. Level 3 is for mission critical systems and those which may have an effect on human safety.

TABLE 3.4-1 SENSITIVITY LEVELS

Automated Information Sensitivity Level	**Explanation** Automated information, automated applications, or computer systems, the inaccuracy, alteration, disclosure, or unavailability of which:
3	Would have an IRREPARABLE impact, permanently violating the integrity of NASA's missions, functions, image, and reputation. Would result in the loss of MAJOR tangible asset(s) or resource(s).
2	Would have an ADVERSE impact, permanently violating the integrity of NASA's missions, functions, image, and reputation. Would result in the loss of SIGNIFICANT tangible asset(s) or resource(s).
1	Would have an MINIMAL impact, permanently violating the integrity of NASA's missions, functions, image, and reputation. Would result in the loss of SOME tangible asset(s) or resource(s).
0	Would have a NEGLIGIBLE impact, permanently violating the integrity of NASA's missions, functions, image, and reputation. Probably would NOT result in the loss of tangible asset(s) or resource(s).

Tier 1 Subcontractors are expected to determine and implement an adequate set of controls so that loss or corruption of development libraries, resources, or assets will not have an adverse effect on delivery schedules, program costs, or flight safety. To facilitate this, the Tier 1 Subcontractor's will utilize existing Corporate or Project practices to provide security and privacy control requirements. For each set of controls being used, the Tier 1 Subcontractor will provide a matrix identifying the extent to which each requirement of JSC 2410.11 Level 1 is being met by the implemented requirements. Any variances will be subject to approval by the Prime. The Prime or NASA Quality Assurance representatives will have the authority to request a review of these practices if a deficiency is suspected, and recommend changes or improvements where such are indicated.

The SVF will meet an operational equivalence of Level 2 and the MBF will provide an operational equivalence to Level 3. During development, both facilities will be controlled and operated as stated above for an equivalent Level 1. Once operational, the MBF will be capable of processing sensitive software applications and data and of promoting them to defined sensitivity levels as appropriate. During the final qualification testing period, the MBF security Level 3 processes and procedures will be demonstrated. Following a NASA assessment, the MBF will be certified as a Level 3 Data Processing Installation (DPI).

3.5 INTERFACE WITH TIER 1 SUBCONTRACTORS

The Prime is responsible for the interface control process between the Prime and the Tier 1 Subcontractors. The basis for coordination of design and data management efforts is the IPT process. AIT/IPTs ensure compatibility within and between end items as a result of continuing communication among all participants. This approach supports early identification of interface issues and a resolution based on a team approach, ensuring a thorough exploration of available options.

The Prime is responsible for directing interface management and for the definition, development, approval, release, and control of interface documentation relating to segment to segment interfaces. Each IPT ensures that interface requirements, as developed, are consistent with product specifications and program physical, functional, and operational requirements. An Interface Control Working Group (ICWG), chaired by a member of the Space Station AIT, integrates interface activities of the products. NASA, the Prime, and Subcontractors maintain membership on the ICWG. The Tier 1 Subcontractor representative on the ICWG is responsible for handling all signature authority for all applicable subcontractors under that Tier. Depending on the quantity and complexity of interfaces involved, a product IPT may form lower-level interface working groups to coordinate interface requirements and develop necessary interface control documentation. Such documentation is subject to approval and control by the ICWG.

The Program ICP establishes a specific process for control and documentation of all physical and functional interfaces of subsystems, equipment, computer software, facilities, installation, and test requirements involving two or more program participants. Interface documentation responsibilities, procedures, and format requirements are also included in the Program ICP.

3.5.1 FLIGHT SOFTWARE INTERFACE CONTROL

Figure 3.5.1-1, ISS Flight Software ICD Process, illustrates the process being used for development of the Prime Flight Software ICDs. The Prime ICDs are composed of two parts: Part 1 defines the functional requirements of the interfaces, and Part 2 defines the implementation of the interfaces. The Part 1s of the Prime ICDs contain the Configuration Item (CI) to CI interfaces. If a CI to CI interface includes a CSCI to CSCI or Firmware Controller interface, the Part 1 will reference one or more of the five Prime Flight Software ICDs (Part 1).

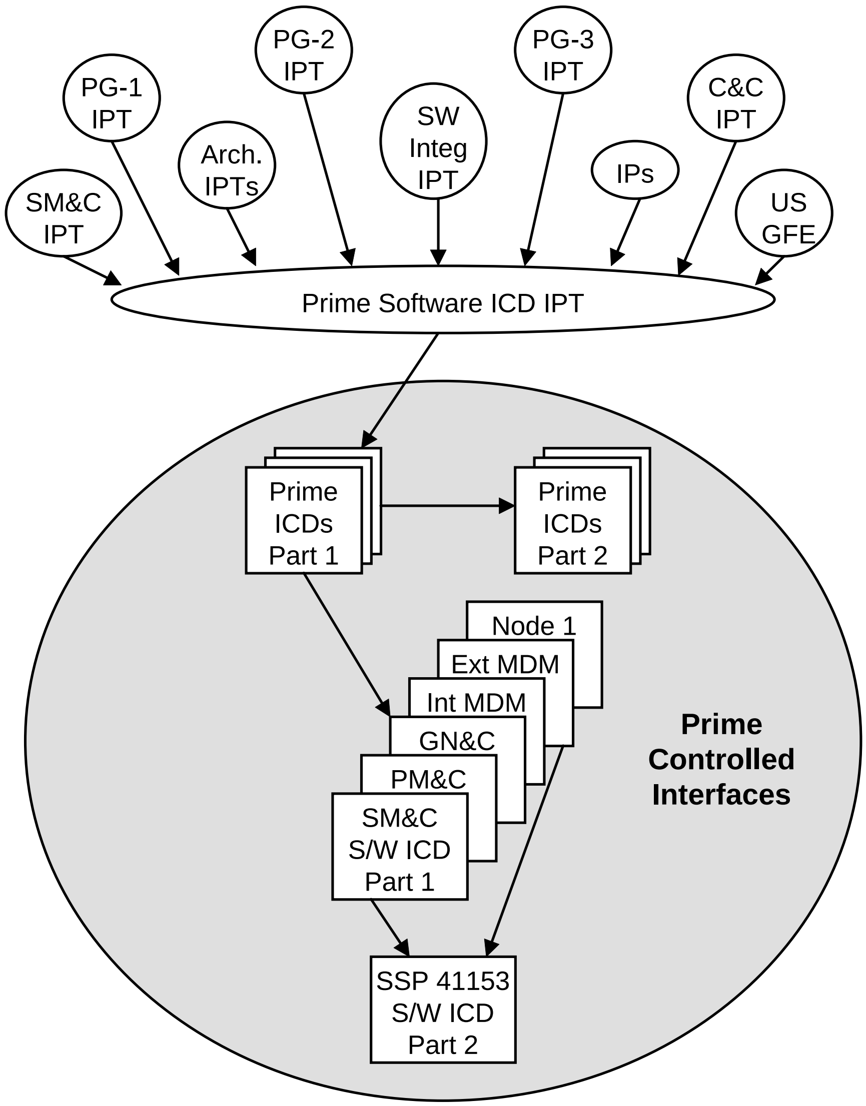


FIGURE 3.5.1-1 ISS FLIGHT SOFTWARE ICD PROCESS

The Prime Flight Software ICD Part 1s are organized by major bus controller CSCIs and contain protocol definitions, standard command definitions and input/output data requirements. Part 2 of the Prime ICDs contain detailed interface requirements for the command, local and user buses that cross end item boundaries. For software interfaces, the Part 2 of the Prime ICDs reference the single Prime Flight Software ICD Part 2, SSP 41153. The Prime Flight Software ICD Part 2 also contains all of the data on the bus (command, local, and user), regardless of whether the data crosses end item boundaries.

The C&DH SWI IPT provides the Flight Software ICD Part 1 section containing the overall interface protocols, timing and sizing requirements, handshaking requirements and Bus operational requirements for all software communicating on the Control or Local Mil-Std-1553 buses. The SM&C IPT will coordinate and provide the external interface information for the crew, ground, space and procedure interfaces.

The primary focal for providing the ICD inputs for the data resident on the Control and each of the Local buses is the developer responsible for the MDM designated as Bus Controller on that bus. Prior to providing the data to the Prime Software ICD IPT, the focal must ensure the following items:

A. Agreement on data from all appropriate Architecture teams;

B. Consistency with the SM&C concepts;

C. Compatibility with the next higher tier ICD;

D. Compatibility with C&DH protocol requirements;

E. Compliance of Program Unique Identifiers (PUIs) with the SSPS;

F. Agreement on the data by both sides of the interface; and

G. Coordination and completeness of interface to GFE and IP products.

The Prime Software ICD IPT will collect all inputs and generate the Prime Flight Software ICDs, which will be reviewed and approved by all affected parties.

Each data item being transferred across the Control and Local buses must be uniquely identified in the Part 1 ICD with a six-character PUI as defined in the Prime SSPS. Each data item, or group of data items, must be associated with a SRS requirement on both sides of the interface. The Part 2 ICD data items are identified by a unique 13-character PUI, also defined in the Prime SSPS, and traced from the Part 1 ICD data.

The C&DH SWI IPT and Software ICD IPT are the primary responsible parties for ensuring total system interface connectivity and correct software functionality. The other Subsystem Provider and Architecture IPTs are responsible for ensuring the interfaces and requirements defined in the ICDs and SRSs provide for adequate implementation of their subsystem’s architecture and higher-level requirements. The Prime Flight Software ICDs will be updated and maintained by the Prime Software ICD IPT.

Each software development group will internally maintain their interface information in the format of their choice.

3.5.2 SVF SIMULATION INTERFACE CONTROL

Figure 3.5.2-1, ISS SVF Simulation Software ICD Process, illustrates the process used for development of the Prime SVF Simulation Software ICDs. The SVF PIDS, S684-10140, describes the functional and performance requirements for SVF simulations provided by the Tier 1 Subcontractors, Prime and IPs.

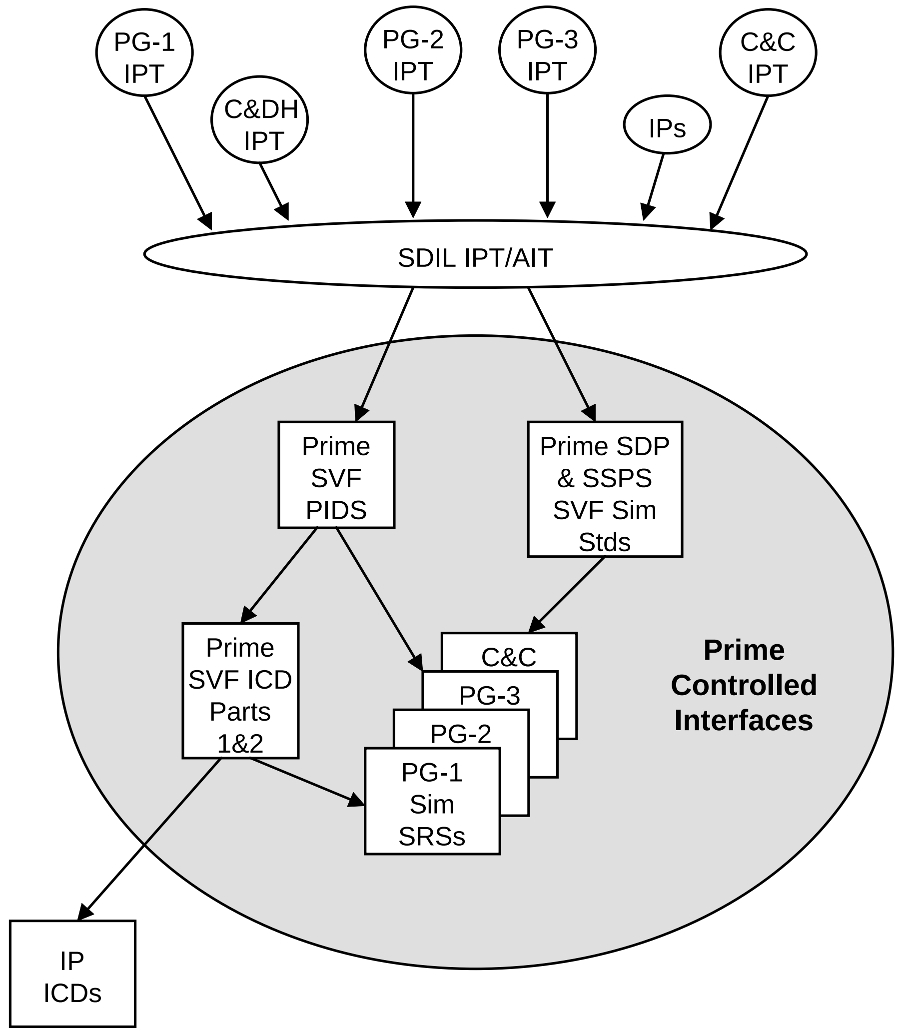


FIGURE 3.5.2-1 ISS SVF SIMULATION SOFTWARE ICD PROCESS

The SVF Simulation ICD, D684-10092-1, controls the interfaces between these simulations. Volume 2 of the Prime SSPS establishes the standards imposed on the simulations to ensure integration of the simulations in the SVF. The Prime SVF Simulation ICD is composed of two parts. Part 1 defines the functional requirements of the interfaces and Part 2 defines the implementation of the interfaces.

The Prime SVF Simulation ICD Part 1 is organized by major bus controller flight software CSCIs and contains protocol definitions, standard command definitions and I/O data requirements. Part 1 & 2 of the Prime SVF Simulation ICD references the Prime and PG flight software ICDs for simulation to flight software interfaces. In addition, Part 1 and 2 contain simulation to simulation interface requirements and interface definition.

Interface requirements contained in the SVF Simulation ICD, as well as functional and performance requirements contained in the SVF PIDS, are allocated to the PG and Prime SRSs. International Partner interface requirements are allocated to the NASA controlled IP ICDs.

3.6 INTERFACE WITH SOFTWARE IV&V AGENTS

The Space Station IV&V agent will perform IV&V as defined in the Interim IV&V Master Plan. All software (including flight and ground) that is identified as high risk/catastrophic/critical will be subjected to IV&V. The software that falls into this category will be identified by the IV&V agent and reviewed with the appropriate ISS program representatives. During the life of the program, additional software may be identified as high-risk due to changing factors such as cost or schedule. This software, when identified, may also be subjected to IV&V at the discretion of the IV&V agent.

3.6.1 INTERFACE BETWEEN IV&V AGENT AND PROGRAM

The IV&V agent will interface formally with the Space Station Program Office through the ISS IV&V Liaison. IV&V will interface informally through participation in JSC, Prime, and Tier 1 Subcontractor IPTs and AITs or their equivalent as non-voting team members. Participation will ensure that the IV&V agent is provided an interface with the Prime and Tier 1 Subcontractors to support informal and formal software development reviews for all life cycle phases. IPT/AIT team leaders will ensure the IV&V agent has the opportunity to participate in team activities as non-voting team members. Participation in remote Tier 1 Subcontractor teams will be accomplished through locally resident IV&V personnel when available. The IV&V agent will assume responsibility for local facility requirements. IV&V representation at each of the Tier 1 Subcontractor sites requesting direct access to the PG will be limited to two personnel.

IV&V requests for additional activities that present potential impacts to program cost or schedule as determined by the Prime or Tier 1 Subcontractors will be coordinated through the C&DH SWI IPT. IV&V requests that the C&DH SWI IPT determines to be outside the scope of the ISS contract will be negotiated between NASA and the Prime contractor. The IV&V agent will make available copies of all technical and issue tracking reports to NASA, the Prime, and the Tier I Subcontractors.

Issues identified by either the IV&V agent or the development contractors will be worked within the appropriate ISS teams (i.e. Prime, Tier 1 Subcontractor, etc.). IV&V technical and issue tracking reports relevant to ISS teams will be transmitted to these teams. Issues which cannot be resolved within these teams will be raised to the C&DH SWI IPT. IV&V representatives and the C&DH SWI IPT co-leads will meet as required to discuss these issues. Issues which cannot be resolved at the C&DH SWI IPT level will be worked through the appropriate higher level ISS program teams, the first being the C&DH IPT, sponsored by the Independent Assessment IV&V Liaison, and then the VAIT, sponsored by the ISS IV&V Liaison. The IV&V agent will ensure that the appropriate working level teams are given sufficient time to respond to IV&V issues and that they are also notified that an issue is being raised to the next level.

3.6.2 IV&V AGENT ACCESS TO DELIVERABLES AND RESOURCES

Deliverable and non-deliverable documentation will be made available to the IV&V agent through participation with Prime and Tier 1 Subcontractor teams responsible for product development. Non-deliverable or informal documentation (i.e. SDFs, SIRDs, etc.) will be made available at the Tier 1 Subcontractors facilities. When available, electronic access to Prime and

Tier 1 Subcontractor's software development documentation will be provided as the information is made available to NASA, Prime, or Tier 1 Subcontractor IPTs/AITs.

Source code documentation will be made available through participation with Tier 1 Subcontractor teams responsible for product development. When available, electronic access to source code will be provided as the information is made available to NASA, SVF, MBF, Prime, and Tier 1 Subcontractors.

Preliminary IV&V documentation assessments will be conducted in coordination with the appropriate level team (i.e. Prime, Tier 1 Subcontractor, etc.) reviews prior to formal reviews. These reviews and the IV&V interaction will be conducted within the general guidelines and in-process review format established for all team members. IV&V will ensure that the assessments are at the appropriate level and within the scope necessary to support the current development phase and software development data maturity.

The IV&V agent will be afforded the opportunity to monitor formal testing of catastrophic/critical/high risk software functions. IV&V will assess the adequacy of the software verification testing for all catastrophic/critical/high risk software functions. IV&V will coordinate with the appropriate verification test teams to resolve test verification deficiencies and will recommend necessary additional testing to ensure thorough test coverage.

IV&V verification recommendations that cannot be implemented through the appropriate test and verification team interfaces will be resolved via the issue resolution mechanism detailed in Paragraph 3.6.1.

The Prime will make the metrics information provided by the Tier 1 Subcontractor status reports available to the IV&V agent.

3.6.3 PARTICIPATION IN REVIEWS AND TESTS

Through participation in the teams, the IV&V agent may participate in all reviews. Upon completion of the IV&V analysis, the IV&V agent will also sign the Certificate of Flight Readiness (CoFR) as required. The IV&V agent will monitor FQT activities relating to the areas of software under IV&V evaluation.

The Tier 1 Subcontractor will provide their IV&V representative at least 30 day electronic or paper mail advance notification of FQT conduct. IV&V will conduct their activities as a passive observer. FQT conduct will not start or stop for IV&V. IV&V questions will be directed to the Test Director, Verification IPT team leader or the C&DH IPT team leader, but not the test conductor.

3.7 SUBCONTRACTOR MANAGEMENT

The Prime is responsible for managing the Tier 1 Subcontractors. The Tier 1 Subcontractors responsible for software development are Boeing D&SG, MDA, and RD. Tier 2 subcontractors are those subcontractors that report to the Tier 1 Subcontractors. The Tier 1 Subcontractors are responsible for implementing the flowdown of this SDP and developing subcontractor SDPs that meet the guidelines established by this SDP. The Tier 1 Subcontractor SDPs will define specifically how the subcontractor plans to develop software in accordance with the processes, standards, procedures, and tools defined by the Prime. Figure 3.7-1, Generic Subcontractor Product Flow, illustrates the generic subcontractor product flow.

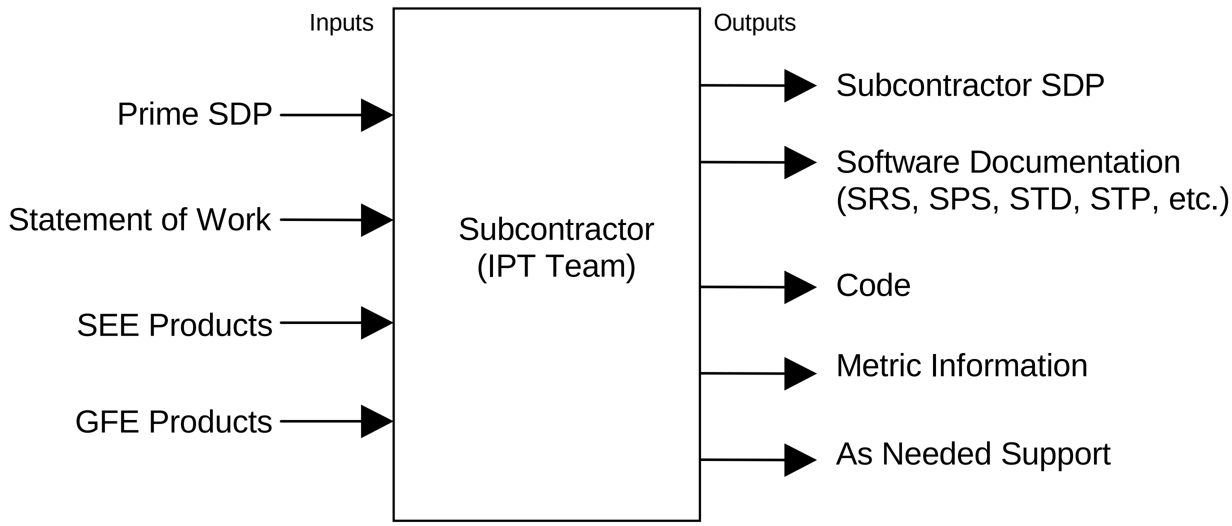


FIGURE 3.7-1 GENERIC SUBCONTRACTOR PRODUCT FLOW

3.7.1 INTERFACE BETWEEN PRIME AND THE TIER 1 SUBCONTRACTOR

Requirements for subcontractor management are defined in the applicable subcontractor SOW and Memorandums of Understanding (MOU) and support the AIT/IPT approach and hierarchy. Subcontractors support the IPT development process as independent IPTs at their facilities and as members of the C&DH Subsystem Provider IPT to ensure horizontal and vertical interfaces. Representatives from the Prime, Tier 1 and Tier 2 Subcontractors are members of the C&DH Subsystem Provider IPT, as appropriate. All software related deliverables (as stated in the Tier 1 SDRLs) are reviewed in the C&DH Subsystem Provider IPT. Subcontractor IPTs at every level of the AIT/IPT hierarchy are expected to uncover any deficiencies in technical quality, cost, schedule, and supportability, based on adherence to the requirements and standards in the SOW, Tier 1 Subcontractor SDPs, system specifications, and quality assurance plans.

3.7.2 SUBCONTRACTOR PRODUCT MANAGEMENT RESPONSIBILITY

Technical system, segment and end item specifications are controlled by the Prime and used by the Tier 1 Subcontractors to develop an SRS for the software product. Any changes to technical requirements are controlled by the configuration management process as required by this SDP and the CMH. The complete set of required technical products varies by contract. The subcontractor is encouraged to prepare cost-effective documentation tailoring consistent with the SDP that meets overall program objectives and life-cycle supportability needs.

3.7.3 SUBCONTRACT EXECUTION

All subcontractors support and participate with the Prime contractor in:

A. Review of subcontractor deliveries against the SDRL, including review for technical accuracy;

B. Establishment and support of Technical Interchange Meetings (TIMs) and reviews related to subcontractor products and technical or performance issues;

C. Development of the Prime and Tier 1 Subcontractor TEPs;

D. Participation at subcontractor Program Management Reviews;

E. Receipt and analysis of routine status information (Quarterly Software Reports by all Tier 1 Subcontractors are provided and include, at a minimum, software metrics information as defined in Appendix B);

F. Participation in developmental prototyping and early software product integration; and

G. Participation in FQT activities and flight testing, as required.

3.7.3.1 SUBCONTRACTOR SOFTWARE ACCEPTANCE

The criteria for conditionally accepting subcontracted software will be the successful completion of FCA/PCA of the Tier 1 Subcontractor's CSCI utilizing the DD1149 process. Final acceptance of software will be successful completion of the end item FCA/PCA utilizing the DD250 process. At FCA, subcontracted software requirements will be shown, through demonstration, analysis, or audit to fulfill system functional capability (or partial capability) from which the requirements were derived. All SDRL items are officially delivered to the Prime contract office and the software executable product is determined to be properly and accurately documented at PCA. The subcontractors will provide support during integration, verification, testing, and end item qualification testing in order to gain full acceptance, as required.

3.8 FORMAL REVIEWS

The program will review the progress of Space Station development at several formal in process reviews. The purpose of the reviews is to provide status of the on-going product development, provide approval for products pertinent to each review and identify any outstanding issues not resolvable prior to the review. These reviews will serve as milestones marking the transition of the engineering products into subsequent phases of the development cycle. Figure 3.8-1, Formal Review Flow, illustrates the associated review with phases in the development cycle.

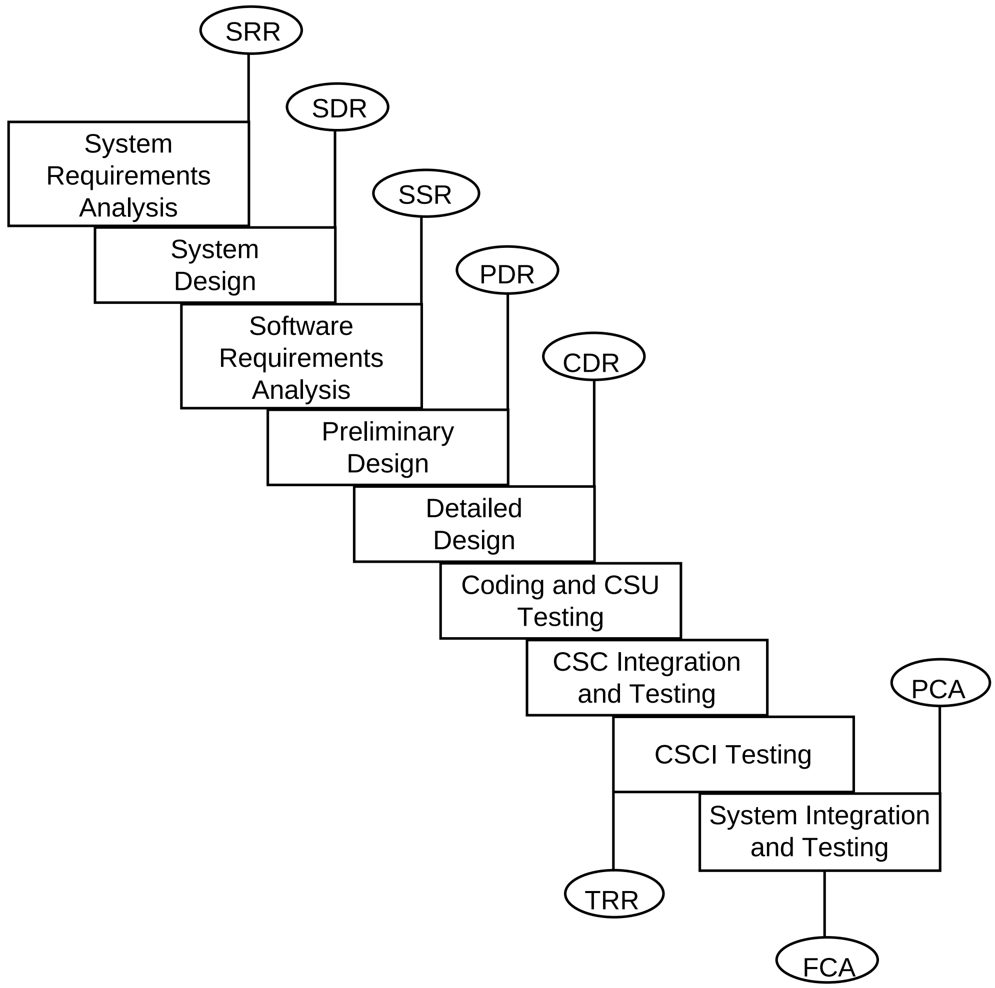


FIGURE 3.8-1 FORMAL REVIEW FLOW

Each review applies a set of criteria to the products of the phase to determine the successful completion of the phase. MIL-STD-1521B will be used as a guide to establish the criteria which are to be applied to each review. Specific criteria for each review will be established prior to the review with concurrence between NASA, the Prime and Tier 1 Subcontractors through approval of the subcontractor's review plan.

Each BSDG is responsible for coordination and tracking of issue generation and resolution of issues resulting from the following reviews of their products. The status and progress of issue tracking will be provided to the Avionics Software Control Board (ASCB) as requested and at specified IPT team meetings. Rev B

Prior to closure of issues, the issue Actionee must coordinate the proposed resolution with the issue originator to obtain agreement on the resolution. Any issue where consensus on the resolution cannot be reached is elevated to the ASCB to be worked. Records of closure of the issues will be maintained in the corresponding SDF.

Figure 3-3, Program Phasing Relationships, identifies the time frame, responsible party and type of baseline for each software document.

3.8.1 REVIEWS HELD FOR NASA (BY THE PRIME)

Two formal reviews will be held for NASA by the Prime. The subcontractors will provide support and participate in these reviews. These reviews include:

A. System Requirements Review (SRR); and

B. System Design Review(SDR).

SRR and SDR are major review milestones for the program. During these reviews, the System Specification, Segment Specifications, Inter-Segment ICDs and SDPs are reviewed by NASA. At the completion of the SDR, the products are entered into the Functional Baseline. The Prime SDP will be reviewed at SRR and updated at SDR. The Prime Flight SDP will be baselined at SDR. The Tier 1 Subcontractor's SDPs will be approved by the C&DH SWI IPT and baselined at SDR plus 75 days. Following this approval, any modifications to the Subcontractor SDPs must have Prime approval.

3.8.2 REVIEWS HELD FOR NASA BY THE BOEING SOFTWARE DEVELOPMENT GROUPS

Rev B

Reviews will be held for NASA by the BSDGs as documented in their individual SDPs. The BSDGs will provide the status of specific review results and recommendation for review disposition to the ASCB. These reviews may take various forms, but will meet the intent of the following standard MIL-STD-1521B reviews:

A. Software Specification Review (SSR);

B. Preliminary Design Review (PDR);

C. Critical Design Review (CDR); and

D. Test Readiness Review (TRR).

Two formal audits will be hosted by the BSDGs and conducted by NASA, these are the:

A. Functional Configuration Audit; and

B. Physical Configuration Audit.

For grandfathered software that requires greater than 20% change, the BSDG responsible for the modification of the software must repeat the previous reviews (even if the review(s) were successfully completed), as well as, implementing and establishing the processes, standards, procedures, and tools as specified in this SDP. (RE: Section 4.2.1.3)

For grandfathered software that requires less than 20% change, the BSDG responsible for the modification of the software must provide a plan for delta reviews, as well as, documenting in section 9 of their respective SDPs any exceptions to the processes, standards, procedures, and tools specified in this SDP. (RE: Section 4.2.1.3) Rev B

3.8.2.1 SOFTWARE SPECIFICATION REVIEW

The purpose of the SSR is to demonstrate to NASA the adequacy of the software and interface requirements to proceed into the design phase. Rev B

Inputs to the SSR are as follows:

A. SSR agenda/plan detailing objective of the review, outline of primary discussion topics and SSR review schedule (required 30 days prior to SSR);

B. Approved PIDS, when applicable;

C. SRS;

D. Preliminary definition of the external interface data elements (input to Software ICD Part 1); and

E. List of all open issues identified from prior reviews and audits.

Successful completion of the following activities is required to determine adequacy of the flight software and interface requirements to proceed into the design phase and thus constitute a successful SSR.

A. Acceptance by review community of the requirements as stated in the SRS including identification of any requirements missing or requiring further refinement and concurrence that the risk to proceed is acceptable;

B. Completion of a functional overview of the CSCI identifying input, processing, and output for all functions;

C. Identification of overall CSCI performance requirements for execution time, storage and sizing, and design constraints;

D. Definition of control and data flow for all functions;

E. Definition of external interface requirements between the CSCI and other configuration items. Preliminary definition of any internal interface requirements;

F. All SRS requirements have been traced back to applicable PIDS and SRS requirements cover PIDS allocation;

G. Qualification requirements identifying methods and levels for verifying each requirement;

H. Applicable quality factors (e.g. correctness, reliability, testability) for each requirement are met;

I. Assurance that the requirements are testable, properly reflect the intent of the higher level specifications (e.g. PIDS) and all applicable higher level requirements have been addressed in the SRS;

J. Traceability between interface information in SRS and ICD Part 1 (PUIs); and

K. Action plan for SSR issue resolution and definition of functional requirements and external interfaces to 95% completion by PDR including assignment of actionee and closure data for each issue.

3.8.2.2 PRELIMINARY DESIGN REVIEW

The purpose of the PDR is to determine if the top-level design of the software is mature and complete enough to advance to the detailed design phase.

Inputs to the PDR are as follows:

A. PDR agenda/plan detailing objective of the review, outline of primary discussion topics and PDR review schedule (required 30 days prior to PDR);

B. Updated SRS, if necessary;

C. Software ICD(s) Part 1;

D. Preliminary definition of the detailed external interface (input to Software ICD Part 2);

E. Preliminary SPS;

F. SDFs;

G. Preliminary STP; and

H. List of all open issues identified from prior reviews and audits.

Successful completion of the following activities is required to determine adequacy of the preliminary design to proceed into the detailed design phase and thus constitute a successful PDR.

A. Acceptance by review community of the requirements as stated in the SRS and confirmation that these constitute approximately 95% of known requirements (i.e., issues and action plans are less than 5%);

B. Functional software structure meets performance requirements for execution time, storage and sizing, and design constraints;

C. Functional software structure meets identified input, processing, and output requirements for all functions and for all states and modes;

D. Definition of control and data flow for all functions;

E. SPS and SDFs together adequately define the basic software design and all CSCI level requirements have been allocated to the design;

F. Traceability of all SRS requirements to the SPS (Software Design Document (SDD) Section 7 equivalent);

G. STP adequately defines formal qualification test plans and test environment;

H. Approximately 80% of detailed external interface data defined with action plans for completing remaining interface definition;

I. Identification of any requirement changes resulting since SSR;

J. Acceptance of the PIDS and Part 1 ICD as the allocated program baseline; and

K. Action plan for PDR issue resolution including assignment of actionee and closure date for each issue.

3.8.2.3 CRITICAL DESIGN REVIEW

The purpose of the CDR is to determine if the detailed design of the software is correct, consistent and complete enough for development to continue to coding and informal testing. This technical review is held to provide a detailed basis for verifying design integrity and compatibility with CSCI requirements and assessment of formal test preparation.

Inputs to the CDR are as follows:

A. CDR agenda/plan detailing objective of the review, outline of primary discussion topics and CDR review schedule (required 30 days prior to CDR);

B. Updates to approved SRS, if necessary;

C. Preliminary Software ICD Part 2;

D. Updated SPS if changes have occurred since PDR;

E. SDFs including detailed design information (SDD);

F. STD Volume 1; and

G. List of all open issues identified from prior reviews and audits.

Successful completion of the following activities is required to determine adequacy of the detailed design to proceed into the coding phase and thus constitute a successful CDR.

A. Acceptance by review community of the requirements as stated in the SRS and confirmation that these constitute approximately 98% of known requirements (i.e., issues and action plans are less than 2%);

B. Acceptance by review community of software design as stated in the SPS/SDF and confirmation that these constitute approximately 95% complete;

C. Detailed design is compatible with functional design structure presented in the SPS;

D. STD Volume 1 adequately documents all test cases necessary to perform formal qualification testing for the CSCI, based on the requirements in the SRS;

E. Detailed design characteristics exist for internal and external interfaces (e.g. range of values, data representation, precision, frequency, etc.);

F. Identification of any requirement changes resulting since PDR;

G. Software design structure meets performance requirements for execution time, storage and sizing, and design constraints;

H. Software design structure meets identified input, processing, and output requirements for all functions and for all states and modes;

I. Traceability of all SRS and software ICD requirements to the SPS (SDD Section 7 equivalent); and

J. Action plan for CDR issue resolution including assignment of actionee and closure date for each issue.

3.8.2.4 TEST READINESS REVIEW

The purpose of the TRR is to ensure that the software test procedures are complete and carry out the intent of the software test plan and descriptions and software to be tested is under formal control and ready for test. This review will be conducted after software test procedures are available and CSC integration testing has been successfully completed.

Inputs to the TRR are as follows:

A. TRR agenda/plan detailing objective of the review, outline of primary discussion topics and TRR review schedule (required 30 days prior to TRR);

B. Updates to the following documents:

    1. Requirements documents (SRS, Software ICD Part 1 and 2)

    2. Design documents (SPS)

    3. Software test documents (STP, STD Volume 1, Test Procedures);

C. SDF assessment results (for documenting of detailed design, handling of known compiler bugs, compliance with coding standards, informal test completion and dry-run results);

D. CSC integration test cases, procedures, and results;

E. Status of test environment including simulation software, test scripts, post processing or data reduction tools, test aids, etc.;

F. Preliminary VDD or equivalent; and

G. List of all open issues identified from prior reviews or audits.

Successful completion of the following criteria is required to determine readiness to begin formal CSCI testing.

A. CSC integration is correct and complete;

B. Formal software test plans, descriptions and procedures are complete and adequate to test requirements;

C. Traceability from all SRS and software ICD requirements to the formal test procedures;

D. CSC integration test cases and procedures are correct and complete and FQT test procedures have been exercised through dry-runs and all known problems have been documented;

E. Identification of all unused code;

F. Determination that the software is maintained under proper configuration control and test environment is ready for use in conduct of formal test; and

G. Action plan for TRR issue resolution including assignment of actionee and closure date for each issue.

3.8.2.5 FUNCTIONAL CONFIGURATION AUDIT

The purpose of the FCA is to validate that a CSCI's actual performance complies with its software and interface requirements specifications defined within the SRS and ICD. Incremental FCAs may be conducted following individual CSCI formal testing and end item Hardware/Software (HW/SW) Integration Testing.

The FCA is conducted using, as a reference, the current revision of Guidelines and Procedures for the Conduct of Functional Configuration Audit (FCA)/Physical Configuration Audit (PCA), D684-10097-01. Rev B

Inputs to the FCA are as follows:

A. FCA agenda/plan detailing objective of the review, outline of primary discussion topics and FCA review schedule (required 30 days prior to FCA);

B. STR;

C. Updates to:

  1. Requirements documents (SRS, Software ICD Parts 1 and 2)

  2. Software test documents (STP, STD Volume 1);

D. Baselined PIDS; and

E. List of all open issues identified from prior reviews and audits.

Successful completion of the following criteria is required to achieve completion of the audit.

A. Complete and correct flowdown of software requirements from the PIDS to the SRS, resulting in authentication of the SRS;

B. Software test plans and descriptions are complete;

C. Traceability from all SRS and Software ICD requirements to the formal test procedures is complete;

D. Software test plans and descriptions provide adequate coverage of all requirements in the SRSs that require formal testing;

E. CSCI verification test results meet the intent of documented test requirements (waivers exist for all cases in which a test did not pass);

F. Confirmation that all PDR, CDR, and TRR open issues are closed or listed as an open action from the FCA; and Rev B

G. Confirmation that all authorized changes to code and documentation have been implemented or listed as an open action for the FCA.

3.8.2.6 PHYSICAL CONFIGURATION AUDIT

The purpose of PCA is to determine if the supporting documentation accurately reflects the qualified as built version of the CSCIs. Upon successful completion of this audit, the product baseline is established. Incremental PCAs may be conducted following individual CSCI formal testing and end item HW/SW Integration Testing.

The PCA is conducted using, as a reference, the current revision of Guidelines and Procedures for the Conduct of Functional Configuration Audit (FCA)/Physical Configuration Audit (PCA), D684-10097-01. Rev B

Inputs to the PCA are as follows:

A. PCA agenda/plan detailing objective of the review, outline of primary discussion topics and PCA review schedule (required 30 days prior to PCA);

B. VDD;

C. All applicable verification documentation (STP, STD, STR);

D. All applicable product specifications and design documentation (SRS, ICDs, SPS, analysis results of SDF); and

E. All applicable all operation/support documents (SUM for user interface software and FSM for FC software).

Successful completion of the following criteria is required to achieve completion of the audit.

A. Final review to confirm correctness and completeness of all operation/support documents and product specifications;

B. Confirm requirements trace between SRS and SPS (SDD Section 7 equivalent);

C. Confirm detailed design (per SDF) matches the code;

D. Confirm adequate information exists in the VDD, SUM and FSM to facilitate control, use and update of the software;

E. Action plans in place to address all FCA open issues/actions, including pending changes to code or documentation previously identified need to be addressed; and Rev B

F. Successful demonstration of recompilation in the MBF.

3.8.3 STAGE INTEGRATION

Rev B

Stage Integration involves the integrated testing of all the CSCIs required for a specific Flight. To support the integrated testing of these CSCIs an Integrated Stage Test Readiness Review (ISTRR) will be held. A discussion of the overall Stage process is found in D684-10020-01, PMI&VP.

3.8.3.1 INTEGRATED STAGE TEST READINESS REVIEW (ISTRR)

The purpose of the ISTRR is to ensure that the stage verification procedures are complete and meet the objectives outlined in the Stage I&V Plan, SVF is ready to support Stage I&V activities, the MBF products are promoted and the SVF is ready and accurately reflects the stage configuration.

Inputs to the ISTRR are as follows:

A. ISTRR agenda/plan detailing objective of the review, outline of primary discussion topics and ISTRR review schedule (required 30 days prior to ISTRR);

B. All software CSCIs have been released through the formal IFL CM Process;

C. Integrated stage software schedules;

D. VDDs for all stage applicable CSCIs;

E. SVF;

F. Stage verification procedures;

G. Status of stage verification dry-run testing;

H. DVO requirements have been allocated to test procedures; and

I. Status of SVF environment including simulations, computing hardware, test scripts, test aids, post processing or data reduction tools, etc.

Successful completion of the following activities is required to determine readiness to begin formal Stage I&V testing.

A. Successful completion of all stage applicable CSCI FQTs;

B. CSCI problem reports for all applicable CSCIs have been identified;

C. Definition of ISS signal data to be processed by the MBF;

D. VDDs adequately define CSCI unique configuration information and include source, executable code and data for each applicable CSCI;

E. Verification test procedures are correct and complete and have been exercised through dry-runs;

F. Integrated stage software schedules representing potential software updates to the stage verification from completion of end item qualification;

G. Determination that all stage applicable software and data has been uploaded to the MBF and is maintained under Prime SCM control;

H. Assurance that the SVF test environment is ready for use in Stage I&V; and Rev B

I. Action plan for ISTRR issue resolution including actionee and closure date for each issue.

3.9 SOFTWARE DEVELOPMENT LIBRARY

All software and documentation under configuration control will be maintained in the Prime SCM library system. The Prime SCM library acts as the Software Development Library (SDL) for the Prime. The SCM library system encompasses the components managed in the Central Data Library (CDL), Central Software Library (CSL), Program Automated Library System (PALS), and Central Program Library (CPL).

The CDL is the central configuration management electronic data repository for ISS ground operational data that is delivered to the Prime for integration, assessment, and configuration control. The data in the CDL is used to supply operational data configurations to the ground facilities. This library is typically thought of as part of the MBF. The CSL is the central software configuration management electronic repository for ISS vehicle flight software, flight data, simulation, and derived load products. This library is typically thought of as part of the MBF. PALS provides for the storage, retrieval, and dissemination of electronic documentation. The CPL is a physical vault to preserve and control all flight software and data deliveries.

The SCM library system is established and maintained by the SCM IPT. Procedures for control of the SCM library system are established by the SCM IPT. The SCM IPT has the responsibility to ensure that established procedures for operation and maintenance of the SCM library system are followed. The CMH will describe how deliveries will be entered, stored, updated, and retrieved and how the content of the library will be protected via access control and security measures.

All flight software and data distributed from NASA for integration or testing will be maintained in the SCM library system. All software delivered by Tier 1 Subcontractors for flight software integration or testing will be processed through the Prime SCM library system where the official status accounting of each delivery will be maintained. Copies of all versions of software and data which have been delivered to the Prime will be maintained in the Prime SCM library system. Figure 3.9-1, Configuration Library Data Flow, depicts the library structure relationship between the Prime and Tier 1 Subcontractor libraries. The CMH defines how the Prime will perform SCM.

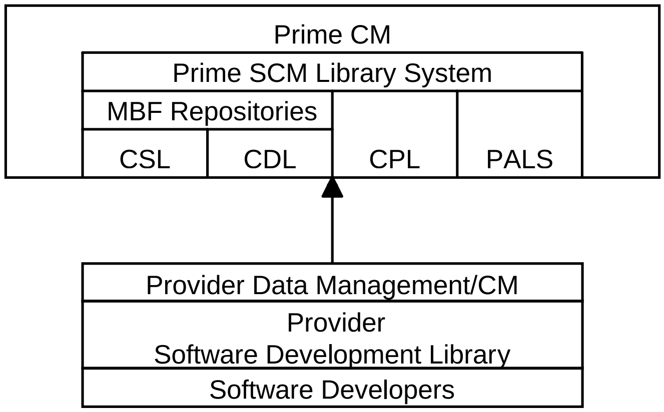


FIGURE 3.9-1 CONFIGURATION LIBRARY STRUCTURE

The SCM process occurs at the various levels of the IPTs (i.e., tiered below the System level along segment lines and the associated segment subcontracting lines). Each subcontractor is required to provide a library process that is delivery compatible either through common tools or common file format with the next higher level library function as mutually agreed between the lower level and higher level contractors.

A local SDL is associated with each of the Subcontractor tier levels. It supports electronic distribution and collection of software and data to and from lower levels and the delivery of documents and application software to the Prime. The local SDL is the central distribution point at a particular level, as well as, the central collection point for promotion to the Prime SCM library system.

Tier 1 Subcontractors will provide their own libraries to support document transfer and software development testing. Deliveries for all major milestones will be provided in the Tier 1 libraries and transferred to the Prime SCM library system.

This approach mandates audit trail information in support of documentation and code deliveries to the Prime SCM library system. The documentation audit trail information consists of:

A. Document title, number, date, source and revision number if applicable;

B. Electronic copy of document;

C. Identification of documentation tool and version used;

D. Identification/description of any other tools used to generate the documentation (e.g. graphics, figures); and

E. Identification of the platform used to develop the documentation.

The software audit trail information consists of:

A. A VDD;

B. Software CSCI/CSC name (or part number) and program unique identifier; and

C. Electronic copy of source code, object code, executable code, compilation script and other files in accordance with the directory structure and file naming conventions described in the SSPS.

3.10 CORRECTIVE ACTION PROCESS

Each Tier 1 Subcontractor may use their own internal software corrective action system which tracks and processes changes to software and software documentation. Each such system will use a Software Problem Report (SPR) or its equivalent form to document inadequate or incorrect statements in program documentation, errors in software design, and/or defects in code. SPRs are submitted to the Tier 1 Subcontractor's SCM which serves as the processing focal point for proposed changes. SCM assigns a number to the SPR, enters it in the SPR log, and routes copies to engineering and SQA for problem analysis and coordination. SCM compiles responses and provides the data to the Tier 1 Subcontractor's C&DH Software Review Board (SRB) for disposition.

The CMH provides an in-depth discussion of the corrective action process used by the Prime. The Tier 1 Subcontractor's internal corrective action process should be defined in their SDP and/or Configuration Management Plan. Problems identified by external interfacing organizations, which impact software under Prime control, are provided to the Prime C&DH SCM IPT for disposition. Figure 3.10-1, Corrective Action Process, depicts the software corrective action process for SPRs coming to the Prime.

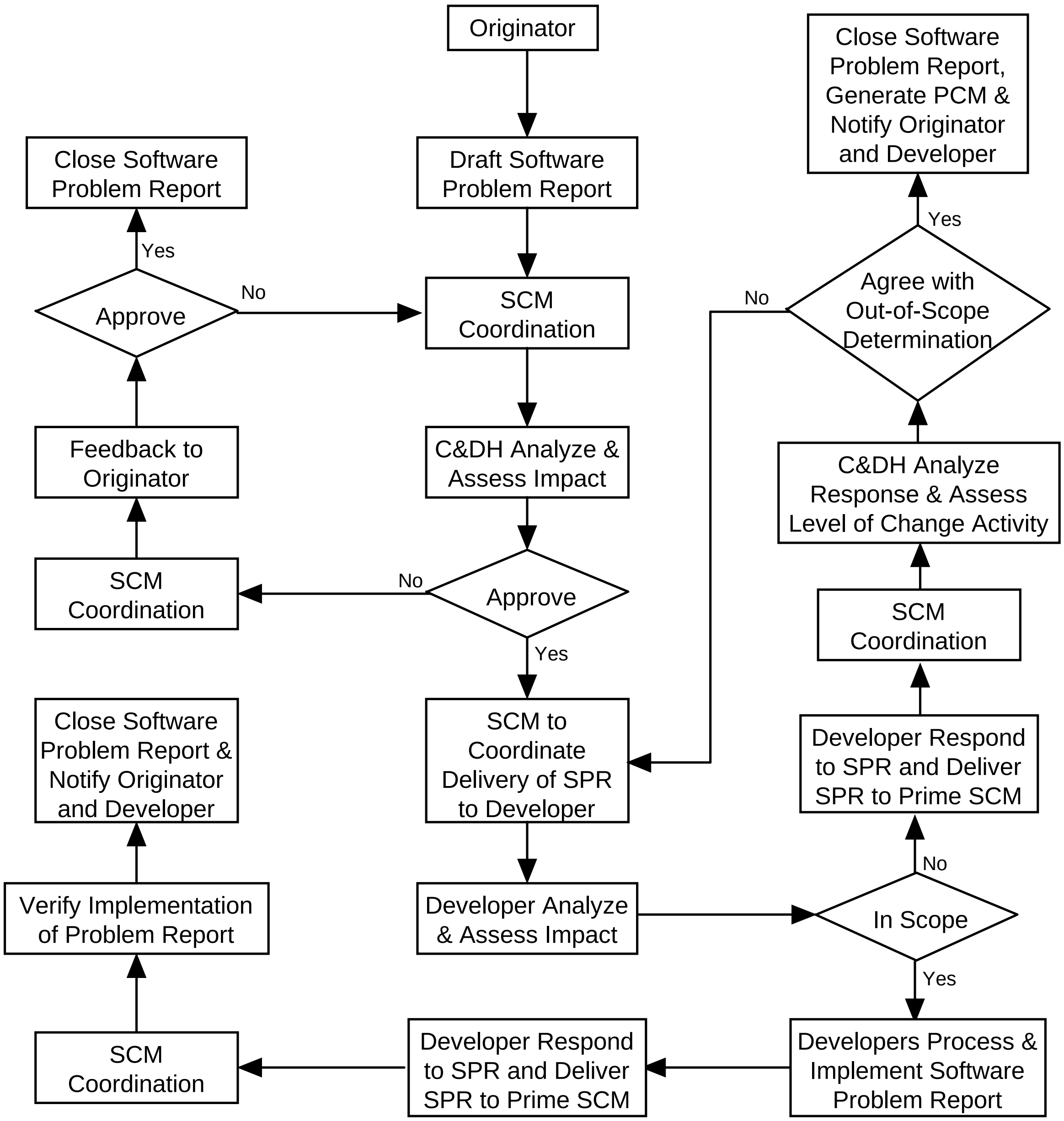


FIGURE 3.10-1 CORRECTIVE ACTION PROCESS

3.11 PROBLEM/CHANGE REPORT

A complete discussion of problem/change reporting forms and processes to be used by the Prime on the ISS is provided in the CMH. For changes with impact to more than one Tier 1 Subcontractor or a product under NASA/Prime Configuration Control, a common Program Change Memo (PCM), as defined in the CMH, should be used for change reporting.

For any change completely internal to a Tier 1 Subcontractor, use of either the PCM or the Tier 1 Subcontractor's own change form is acceptable. The format used for problem identification/ reporting may be in subcontractor format, provided any problem resulting in a change follows the guidelines previously specified for the use of the PCM. Each Tier 1 Subcontractor will document in their SDP the reporting mechanism, including format and data content, to be used for software problems. Data content should contain, at a minimum, the information specified in paragraph 10.2.5.11 of the SDP DID, DI-MCCR-80030A. Figure 3.11-1, ISS Software Problem Report, contains a sample software problem report form.

PROBLEM REPORT ENTRY PR #: _______________ SHEET 1 OF 2

CONTRACT/ECP: ____________________ SYSTEM/CI/CSCI: _______________ DATE: ____________
TITLE ____
PHASE: ___________

ORIGINATOR: _____________________________ IPT NAME: ___________________________________
TELEPHONE: _____________________
CATEGORY: CODE ______ DOCUMENTATION ______ DESIGN ______ HARDWARE _____
PRIORITY: 1 __ 2 __ 3 __ 4 __ 5 __ NEED DATE: ________________
1 - MANDATORY CHANGE 2- PERFORMANCE DEGRADATION 3 - XXXXXXXXXXX
4 - XXXXXXXXXXX 5 - XXXXXXXXXXX

CONFIGURATION OF ITEMS REQUIRING CHANGE
HARDWARE ID: ________________________________ SERIAL NO. ___________
SOFTWARE ID: ________________________________ VERSION ___________
DOCUMENT ID: ________________________________ REVISION ___________
TEST ID: ________________________________

REASON FOR CHANGE: (ATTACH SHEET IF NECESSARY)

ANALYSIS INFORMATION
ANALYST: ____________________________ IPT NAME: ___________________________________
PHONE: ______________ ASSIGN. DATE: _________________ COMPLETE DATE: _______________
MANHOURS EXPENDED: ___________________________
DESIGN DEFICIENCY ___ PRODUCT IMPROVEMENT ___ OTHER ___ (CHECK ALL THAT APPLY)

RECOMMENDED CORRECTIVE ACTION: (ATTACH SHEET IF NECESSARY)

IMPACT OF RECOMMENDED CORRECTIVE ACTION:
SCHEDULE IMPACT: YES ____ NO ____ ESTIMATED MANHOURS TO COMPLETE: _____________
INTERFACE IMPACT: ____

APPROVED _______ DISAPPROVED _______ DEFER _________ DATE: _________________________
PLANNED VERSION INCORPORATION: ______________ BLOCK INCORPORATION _____________

IPT LEADER (SIGNATURE): _______________________ PRIORITY BY IPT: _______________________

FIGURE 3.11-1 ISS SOFTWARE REPORT

PROBLEM REPORT ENTRY PR #: ________________ SHEET 2 OF 2

CORRECTIVE ACTION TAKEN: (ATTACH SHEET IF NECESSARY)

CSU(S): CHANGED ___________ ___________ ___________ ___________ ___________
ADDED ___________ ___________ ___________ ___________ ___________
DELETED ___________ ___________ ___________ ___________ ___________

CONFIGURATION OF ITEMS AFTER CHANGE IMPLEMENTATION:
HARDWARE ID: ____ SERIAL NO: ______________________
SOFTWARE ID: ____ VERSION: ______________________
DOCUMENT ID: ____ REVISION: ______________________
TEST ID: ____

PERFORMED BY: ________________________ IPT: __________________ TEL. #: ________________________
DATE COMPLETED: ______________________ MANHOURS EXPENDED: ___________________________

VERIFIED BY: ________________________ IPT: __________________ TEL. #: ___________________________
DATE COMPLETED: ______________________ MANHOURS EXPENDED: ___________________________

VERIFICATION REMARKS: (ATTACH SHEET IF NECESSARY)

CLOSEOUT REMARKS: (ATTACH SHEET IF NECESSARY)

DISTRIBUTION:
NAME: _______________________________ IPT: _____________________________ M/S: ________
NAME: _______________________________ IPT: _____________________________ M/S: ________
NAME: _______________________________ IPT: _____________________________ M/S: ________
NAME: _______________________________ IPT: _____________________________ M/S: ________
NAME: _______________________________ IPT: _____________________________ M/S: ________
NAME: _______________________________ IPT: _____________________________ M/S: ________

CLOSEOUT BY: __________________________ DATE: ____________ TEL #: ___________________________
(SOFTWARE CONFIGURATION MANAGEMENT)

FIGURE 3.11-1 ISS SOFTWARE REPORT (Concluded)

3.12 SOFTWARE METRICS MANAGEMENT

3.12.1 ORGANIZATION AND RESOURCES

Software metric reporting is accomplished by the Tier 1 Subcontractors who are responsible for the software development. Software metrics are reported to the C&DH SWI IPT who, in turn, summarizes and analyzes the data and reports the results in the form of a Software Metrics Status report to the C&DH Subsystem Provider IPT and subordinate teams. This data is also included in the Prime Program Metrics Database.

3.12.2 PURPOSE AND SCOPE

The goals of software metrics are to provide management visibility into the software development process and to promote the timely development of quality software products. The analysis and assessment of software metrics provides early warning signals to highlight potential system development, test, integration, or schedule problems before they become unrecoverable, causing schedule disconnects and/or slippage.

3.12.3 SOFTWARE METRIC REPORTING METHODOLOGY

Software metrics are reported at the CSCI and major CSC level (as identified in Appendix A). They are reported quarterly to the C&DH SWI IPT in a spreadsheet format. Software metrics include plans, estimates, and actuals for the reporting period. Reporting procedures are detailed in Appendix B of this document.

3.12.4 SOFTWARE METRICS

The following software metrics, or software management indicators, are reported. Detailed descriptions of each metric are included in Appendix B.

A. General data including reporting period, CSCI/CSC name, CSCI/CSC manager, etc.;

B. Software size;

C. Design, code and informal test progress;

D. Software volatility;

E. Formal test progress; and

F. Computer resource utilization.

3.13 FLIGHT SOFTWARE BUILD

The flight software build process described in this document refers to those activities resulting from the delivery of Tier 1 Subcontractor software and data acceptance packages to the MBF used in the preparation of the flight software and data products for delivery to the program integration, verification, training and operations facilities. This build applies only to on-orbit down-loadable software. Flight Software Builds created for the purposes of integration, verification, assessment and operations are produced as products of the MBF. Tier 1 Subcontractors may use the MBF to produce informal builds as preparation for their PCA acceptance. The first formal products generated from the MBF will be used by the Prime for horizontal testing in the SVF. In the software life-cycle, build process responsibilities occur after development is complete and continue for production and SCM support purposes for the remaining life of each CSCI. Software builds for testing of development software and data configurations prior to formal delivery to the MBF are the responsibility of the development organization.

3.13.1 TIER 1 RESPONSIBILITIES

Tier 1 Subcontractors are responsible for releasing contract deliverable CSCIs to the Prime SCM library system according to the standards in the Space Station Data Management Plan, D684-10002-1, and the appropriate contract end item agreements. The CSCI acceptance packages will contain the items as listed in SDS SS-VE-035 of the Tier 1 Subcontractor’s contracts (e.g., source code, object code, data tables, version descriptions and loader instructions for creating an executable of the CSCI from the delivered source). These acceptance packages will be delivered using directory structures and file naming conventions as specified in the SSPS.

3.13.2 PRIME CONTRACTOR RESPONSIBILITIES

The Prime is responsible for the activities performed at the MBF to support SCM and the software build. These include:

A. Acceptance - Each delivery package is received by the MBF and inspected to validate the completeness and accuracy of the package components. Activities conducted as part of the inspection process include:

   (1) Ensure all components of the delivery in the Software Development Integration Lab (SDIL) SQA release directory are present by verifying the completeness and accuracy of the CSCI delivery package against the SQA release notification;

   (2) Initial deliveries of a CSCI to the MBF will be compiled using the program approved compiler and linker version as defined in the VDD. The compiled CSCI is then bit-for-bit compared to the executable provided with the original delivery package. This process will also be performed on subsequent releases to the MBF where a build script is modified, or there is a change in the compiler or linker version as defined in the VDD;

   (3) Data-only deliveries are validated per the SSPS to ensure correctness prior to their incorporation into the baselined program database;

   (4) The CSCI(s) and associated data are then available to the SVF for informal checkout of the load. Notification will be provided to the SCM IPT upon successful completion of this testing; and

(5) Approved CSCI(s) and data-only deliveries will be subjected to a security inspection. Upon successful analysis, the software will be promoted to AIS Security Level 3 and archived.

B. Logging and Check-in - After the successful bit-for-bit comparison, testing, and promotion, the delivery package is now under formal Prime Configuration Management (CM) control. The package is logged and the software and data objects are formally checked into the proper library repository which is part of the Prime SCM library system. Status notifications are prepared and distributed to the appropriate participant and management organizations.

C. Build Distribution - Formal CM MDM executable images are placed on the MBF standard output server by the SCM IPT. Ground facility customers for individual MDM builds, or stage groupings of MDM builds, will log on to that server to retrieve the images.

3.13.3 FLIGHT SOFTWARE BUILD CREATION

Flight Software Builds for use on-orbit are provided to the launch site by the end item provider following successful MBF acceptance, logging and check-in. Flight Software Builds for use in comparing MDM executables are created using the CM and Build Tools resources of the MBF. These MBF created Builds are discarded upon successful logging and check-in of the delivered products. Copies of accepted executable images from development sources are passed to the requesting facility. See Figure 3.13.3-1, MBF Build Process Flow. MBF provided Builds come only from software and data formally delivered by Tier 1 Subcontractors.

3.13.3.1 INTEGRATED FLIGHT LOAD

An Integrated Flight Load (IFL) identifies an integrated Flight Software and Data Subsystem configuration for a particular Flight Stage. IFLs are used for formal integrated testing and for Flight. The details of the IFL definition, build and distribution process will be defined in the Software Configuration Handbook, D684-10293-01.

Flight Software Builds may be grouped together for the purpose of a specific flight or a particular test. The grouping of Builds is known as an Integrated Flight Load. For each IFL the Prime will generate the necessary VDDs defining its contents.

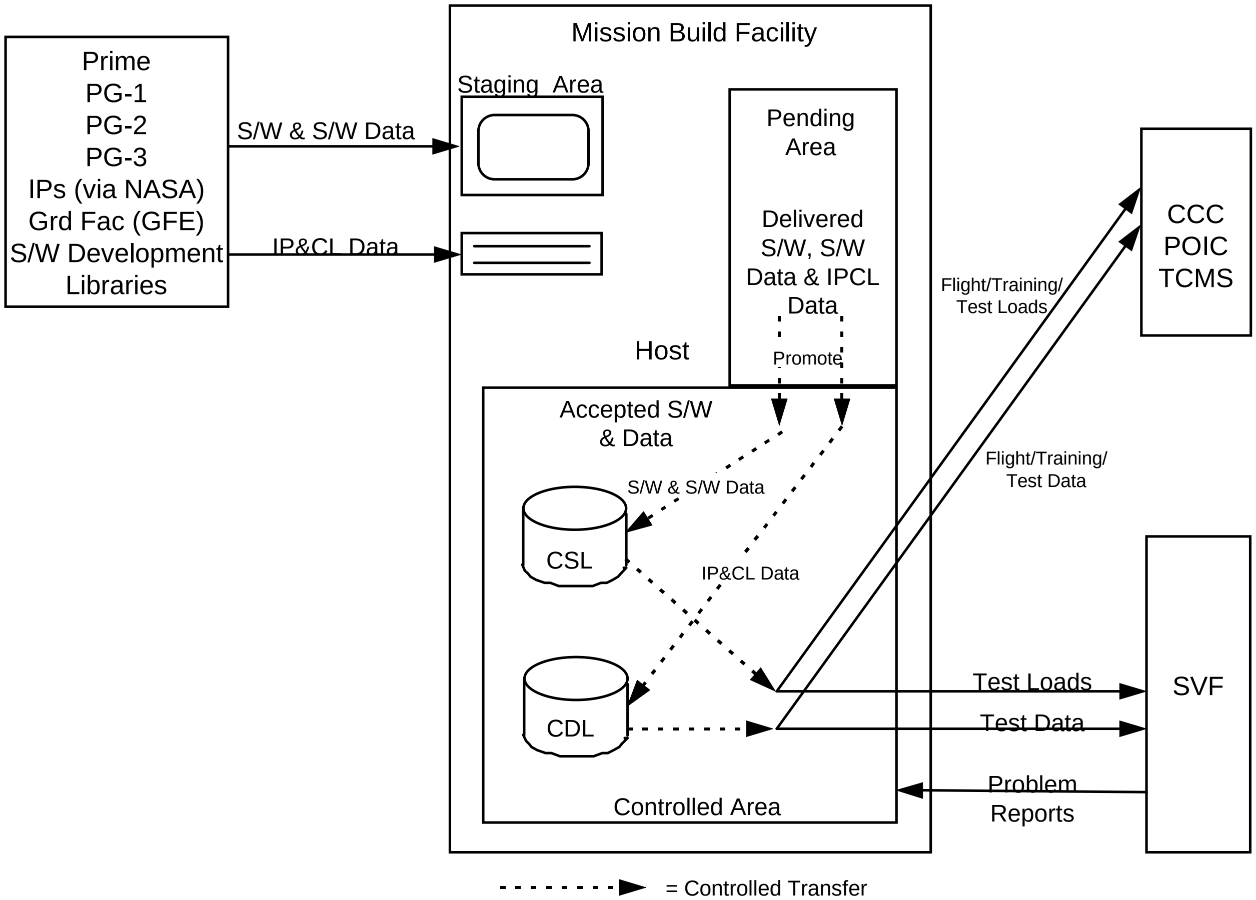


FIGURE 3.13.3-1 MBF BUILD PROCESS FLOW

3.14 FIRMWARE MANAGEMENT

The proper documentation of firmware is essential to both development and sustaining engineering activities. The criteria in this section should be used by each subcontractor to ensure consistent identification of the FC and whether the firmware is to be developed and documented as software or hardware. The life-cycle followed for firmware management depends upon the complexity of the FC.

3.14.1 CLASSIFICATION OF FIRMWARE

The following definitions will be used to differentiate what items are classified as programmable hardware versus firmware.

A programmable hardware device is an integrated circuit whose function is defined by a configuration program which defines the interconnections and functions of the programmable logic elements within the hardware. The configuration program is typically stored in memory circuits which may be part of the same programmable hardware device, or in Read Only Memory (ROM), Programmable Read Only Memory (PROM), Erasable Programmable Read Only

Memory (EPROM), Electrically Erasable Programmable Read Only Memory (EEPROM) and sometimes Random Access Memory (RAM). Typical programmable hardware devices are:

A. Programmable Logic Device;

B. Gate Array; and

C. Field Programmable Gate Array.

Firmware is defined as computer programs stored in a hardware non-volatile silicon-based memory device (ROM, PROM, EPROM, EEPROM). A computer program is a sequence of coded instructions which encode a thought process or algorithm that may be executed by a computer system.

Once the firmware designation has been made, an analysis must be performed to determine whether the FC is considered simple or complex. This is accomplished through a review of the end item specification requirements for allocation to a FC. FCs determined to have major C&DH system interfaces are designated complex FCs, included in the flight CSCI list in Appendix A of this plan, and should follow the software life-cycle defined in Section 4.2. A major C&DH system interface is one where the FC is tied directly to a C&DH system resource (MDM or network) and the interface consists of commands and/or status beyond the on-off/start-stop basic commands sent and simple status returned.

All functionality designated as simple follows the hardware life-cycle and has all requirements documented in B2 specifications. ROM electronic and electrical engineering part numbers will serve as the configuration managed identifier of firmware after it becomes part of the hardware.

A request must be submitted to the C&DH SWI IPT and/or documented in the Tier 1 Subcontractor's SDPs for any functionality classified as hardware or simple FC based on the above criteria. Approval by the C&DH SWI IPT will be authority to produce and document the product according to a hardware life-cycle.

3.14.2 FIRMWARE DOCUMENTATION

Complex FCs will be documented following the tailored DOD-STD-2167A standards previously identified for flight software in Section 3.0. Firmware Support Manuals, which contain all necessary maintenance and modification information, will be provided by the subcontractor for both complex and simple FCs. FCs containing Commercial-Off-The-Shelf (COTS) products will be handled in accordance with Section 4.3. Hardware/FC interface simulations will be provided with the FCs, or their software, to the SVF for testing during horizontal testing.

3.15 SHARED MDM INTEGRATION STRATEGY

A shared MDM is any MDM that contains application software developed by two or more product developers. Each processor may have its own executive based on the decision of each BSDG. Shared MDM software integration requires that special software development and test processes be put in place to ensure the correct integration of the MDM software. This strategy applies for both standard and enhanced MDMs. Rev B

In addition to software sharing of a MDM, the MDM software may be resident in a MDM which is part of a launch package or end item that is the responsibility of a different BSDG (e.g., the Power Management and Control Application (PMCA) CSCI developed by B-CP resides in the Lab which is a B-HSV end item). Therefore, the integration of a MDM may involve several parties including the: Rev B

A. MDM Owner who is responsible for the end item delivery for Launch;

B. CSCI Owner who is responsible for the integration of the CSCI within the specific MDM; and

C. CSC Supplier who supplies a software package as a CSC to the CSCI owner.

Appendix A provides a list of CSCIs and MDMs with their associated owners. For shared MDMs, the list also includes major CSCs being provided by a developer other than the CSCI owner.

3.15.1 PROCESSES AND RULES FOR INTEGRATION

Following the processes and rules in this section will minimize integration issues for end item deliveries. These processes will be coordinated by the C&DH SWI IPT which has representation from all of the Tier 1 Subcontractors on the ISS program.

A. MDM Owner

The Tier 1 Subcontractor responsible for delivering the MDM for flight is designated as the “MDM Owner” for that specific MDM. The MDM Owner is responsible for:

(1) Completion of the analysis and documentation of all required activities to ensure “Flight Readiness Certification” of the launch package which contains the MDM as or within an end item;

(2) Completion of the analysis and documentation of all required activities to ensure “Flight Readiness” of the software, including the contents of the EEPROM containing the CSCI;

(3) Performing end item testing of the MDM or end item containing the MDM; and

(4) Establishing an overall end item development and integration schedule.

B. CSCI Owner

The Tier 1 Subcontractor responsible for delivering the CSCI load to the MDM Owner is designated as the “CSCI Owner” of that MDM. The CSCI Owner is responsible for:

(1) Establishing early-on with the CSC supplier, the acceptance test criteria and supporting test bed configuration identification and development;

(2) Acquisition of all CSCs from all pertinent product developers in a timely manner;

(3) Compilation, integration and test of all software packages in the MDM as one CSCI;

(4) Supporting the MDM Owner in the CSCI loading and ”Flight Readiness Certification” processes;

(5) Performing CSCI FQT as defined in Section 5 of this SDP;

(6) Supporting the MDM Owner in performing the end item testing;

(7) All CSCI documentation and reviews;

(8) Delivering the CSCI to the MBF;

(9) Meeting the schedules agreed to with the MDM Owner for the end item deliveries and Prime for stage software deliveries;

(10) Allocating and managing the CSCI’s computer resources; and

(11) Performing configuration status accounting for each deliverable CSCI.

C. CSC Supplier

A product developer responsible for delivering a software package as a CSC to a CSCI Owner is designated as a “CSC Supplier” for the associated MDM. The CSC Supplier is responsible for:

(1) Configuring the CSC delivered with the necessary data loads. The CSC supplier is the data owner for all data associated with the delivered software;

(2) Support, review and concur that the CSCI Owner has correctly integrated the CSC in the CSCI build process, as part of the CSCI integration activity;

(3) Performing unit and CSC level testing;

(4) Performing informal CSC/MDM executive integration level testing;

(5) Providing MATE simulations to the CSCI Owner;

(6) Supporting the CSCI Owner in performing the FQT;

(7) Supporting the MDM Owner in performing the end item test;

(8) Participation in the program activities in support of both the CSCI Owner and MDM Owner to ensure “Flight Readiness Certification” of the specific MDM software capability and correctness of the EEPROM load;

(9) Providing material for the CSCI SRS, SPS, STD, SDF and other documents, as appropriate, in a timely manner;

(10) Participating in the CSCI reviews;

(11) Supporting the CSCI owner in computer resource utilization; and

(12) Meeting schedule agreed to with the CSCI Owner.

The Ada tasking structure as defined in the C&DH Architecture Notebook, D684-10500, will be followed by all software developers. The software developers participate with the SWI IPT in developing the architecture described in the C&DH Architecture Notebook.

4 SOFTWARE ENGINEERING

4.1 ORGANIZATION AND RESOURCES - SOFTWARE ENGINEERING

4.1.1 ORGANIZATIONAL STRUCTURE - SOFTWARE ENGINEERING

Software engineering at the Prime will be the responsibility of the C&DH SWI IPT and its sub-teams. The SWI IPT is comprised of the Software Systems Engineering Group, PG/Team Management Group and Software Integration Group as shown in Figure 4.1.1-1, Software Integration IPT Organization. In addition to the teams reporting directly to the SWI IPT, the C&C Software IPT and the Software teams at each of the three PGs have an indirect link to the SWI IPT for development of software products, progress reporting and support to software integration activities.

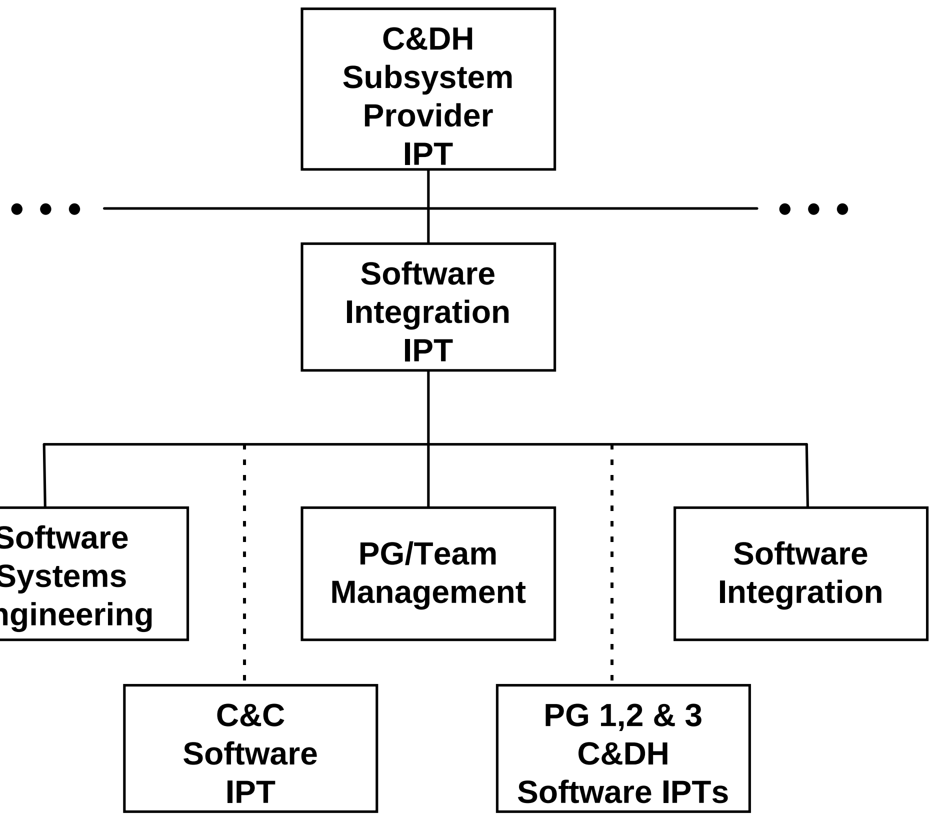


FIGURE 4.1.1-1 SOFTWARE INTEGRATION IPT ORGANIZATION

The SWI IPT is jointly lead by a Boeing and NASA leader and meets on a regular basis to coordinate plans and schedules, provide status, review metrics and resource utilization, distribute information and discuss/resolve issues and risks. These co-leads have approval authority for all Tier 1 Subcontractor deliverables as defined in the contract SDRLs and all tailoring of the Prime SDP and SSPS as identified in Section 9 of the Tier 1 Subcontractor SDPs. Details of the SWI IPT organization may be found in the SWI IPT TEP.

The SWI IPT's role is to ensure communication and design consistency between the four development teams and perform software integration of their individual pieces into a total software product. The SWI IPT also interfaces to the C&DH Subsystem AIT to coordinate software issues, schedules and technical decisions with the IPs.

The Software Systems Engineering Group has the responsibility for development of the overall C&DH software protocol and control concepts, definition of adaptation and reconfiguration data, support of software safety assessments and definition of software interfaces particularly for IP, GFE and FGB software.

The PG/Team Management Group has the responsibility for definition of software standards and methodologies for all software developers, assessment of development progress and compliance with program standards, review and approval of subcontractor deliverables, collection and reporting of C&DH software metrics, conduct of software stage reviews and development and maintenance of integrated software schedules.

The Software Integration Group has the responsibility for supporting and monitoring the HI MDM Utilities development for the Prime, stage software integration planning and implementation, coordination of Ada Flight Software Development Environment issues, and providing support to C&DH Verification IPT and SVF developers for definition of test facility usage requirements.

4.1.2 PERSONNEL - SOFTWARE ENGINEERING

This section describes the C&DH SWI IPT personnel requirements. Due to the diverse areas to be covered by the C&DH SWI IPT, a wide range of personnel skills are required by this team. Membership on the team will consist of:

A. Boeing team leader with extensive software development and verification experience and strong management capabilities;

B. NASA team leader with extensive software development and verification experience and strong management capabilities;

C. Boeing, NASA and Prime subcontractor personnel, experienced in software development, software verification, program planning and scheduling and subcontract management/procurement;

D. IV&V representative to coordinate IV&V software issues and activities;

E. Representatives from each of the Tier 1 Subcontractors for integration, verification, interface control definition, activity network development and overall planning functions;

F. Risk Management representative;

G. SCM representatives; and

H. S&MA representatives.

A list of the SWI IPT personnel, including the discipline they are representing, is kept current in the SWI IPT TEP.

4.1.3 SOFTWARE ENGINEERING ENVIRONMENT

The SEE comprises the U.S. portion of the Space Station Program resources (hardware, software, firmware) used for development, integration, testing, verification, build, and management of the flight software; including integration, build, and management of the flight data and flight related operational ground data. Continuous assessment of the software life cycle management requirements is used to determine the proper collection of support tools and management processes which are specified, provided, or supported by the SEE. The Prime SEE tools will be used for the Prime Mission Build capability (see Section 3.13 of this plan).

A subset of the SEE tools set will be defined and managed by B-HOU for BSDG use. These tools are provided to the Tier 1 Subcontractors in cases where the C&DH Subsystem Provider IPT mandates a particular tool for program-wide use. The Aonix Alsys AdaWorld VAX/VMS to 80386 Cross compiler, including optimizer, debugger and linker packages and the corresponding problem reporting compilers, are the only mandated SEE products. These mandated products are only mandated for the production of Boeing ISS Flight Software running on MDMs. The Prime will require that delivered software products be compiled using these standard SEE products, a standard operating system version, and standard compiler and linker options. All exceptions to this requirement must be coordinated and approved by the C&DH SWI IPT. The use of SEE provided tools does not otherwise restrict or preclude the use of any non-SEE hardware or software tools in any developer or program facility. Rev B

4.1.3.1 SOFTWARE ITEMS

Types of software tools supported by the Prime SEE include: software design support tools, software implementation support tools, software testing tools, test support and management tools, documentation tools, data transfer and conversion tools, build tools, and configuration and program management tools. The SEE configuration list, included in Appendix C, is maintained by the C&DH SWI IPT and is a list of SEE tools.

4.1.3.2 HARDWARE AND FIRMWARE ITEMS

Hardware and firmware items provided and/or supported by the SEE are determined by the functional requirements of the SEE as detailed in the implementation specifications.

4.1.3.3 PROPRIETARY NATURE AND GOVERNMENT RIGHTS

Hardware and software purchased from commercial vendors and provided to development organizations or program facilities will be used according to the appropriate restricted access rights established by the particular licensing agreement.

The software, hardware and firmware items defined in this SDP are furnished to the government in accordance with applicable contract provisions.

4.1.3.4 INSTALLATION, CONTROL, AND MAINTENANCE

The C&DH SWI IPT will provide maintenance and on-going support for all Prime SEE tools. Installation and checkout of SEE tools for use in the MBF/SVF will be provided by PG-1. The mandated subset of SEE tools used by the Tier 1 Subcontractors will be controlled by the Prime, but installed and maintained by the Tier 1 Subcontractors.

The Prime will evaluate mandated software tools periodically (3 to 4 years) to determine if the tool should continue to be used, and possibly frozen for the duration of the program, or replaced. Any mandated software tool that is retired by the vendor will be evaluated for continued use within the program.

4.2 SOFTWARE STANDARDS AND PROCEDURES

The following subsections describe the techniques and methodologies which will be employed in the software development process, the design and coding standards, and the software development folders used to capture the in-process status and development information.

4.2.1 SOFTWARE DEVELOPMENT TECHNIQUES AND METHODOLOGIES

The purpose of this section is to define the Software Development methodology to be applied to the software products covered by this SDP. This methodology will meet the requirements of DOD-STD-2167A and will assure the software product meets system requirements early in the development process by modifying the traditional waterfall life cycle to include prototyping and early brassboard checkout of high program risk functions. A side benefit of this life cycle tailoring is the option to use prototype or brassboard software to become product code, once upgraded to meet program standards. This section will describe the software development processes using the SEE environment to be used from requirements analysis through FQT. Each Tier I Subcontractor will define in their SDP, all detailed techniques, methods and tools consistent with this SDP. In this section the term methods, methodologies, and techniques refers to the life cycle approach or development steps and also to the means used in the design of the software.

4.2.1.1 CATEGORIZATION OF SOFTWARE DEVELOPMENT

Each Tier I Subcontractor will specify the method by which their software is categorized, designated as either application or developmental support software. Developmental support software does not become part of a deliverable CSCI. Examples of developmental support software are test stubs, prototype unique software, compiler aids, test support equipment, etc. Developmental support software is documented in the SDFs and is reviewed by the Prime as needed. Categorization of software as developmental support will be documented in the Tier 1 Subcontractor's SDPs.

Deliverable software is provided to the Prime per the delivery schedule included in the CFEL and DIL in each Tier 1 Subcontractor's contract. These products will be configuration controlled at the Tier 1 Subcontractor until FCA/PCA, at which time the configuration control transfers to the Prime SCM library system. Application software documentation is formally reviewed by the Prime throughout the development life cycle.

Non-deliverable software is configuration controlled within the Tier 1 Subcontractor organization. It is available for access at the Tier 1 Subcontractor facility.

4.2.1.2 REQUIRED DEVELOPMENT STEPS

The ISS Program will provide for a Software Development methodology for all software categorized as application, as described in this section. Details of required documentation for Software Development is described in Section 3. Figure 4.2.1.2-1, Software Development Process, illustrates the mandatory processes and milestones required by this SDP. Table 4.2.1.2-1, Detailed Development Process, provides a detailed overview of the Software Development process inputs, goals, SDF contents and outputs. The Tier 1 Subcontractor SDPs should also include development environment, participants, CM tools and metric categories. Table 4.2.1.2-2, ISS Software Types vs. Mandatory Development Steps, provides a matrix of the types of ISS software and the applicability to these mandatory steps. The mandatory items are detailed in the following subparagraphs together with additional steps whose intent should be met.

It is expected that detailed process guidelines and compliance checklists will be created by the Tier I Subcontractor to confirm compliance with the requirements of this plan. These guidelines and checklists will be documented in internal Tier 1 Subcontractor documentation such as an SSPS, software developer's handbook, operating procedures, etc.   It is expected that the Requirements developers, Software Quality and Software Safety disciplines are heavily involved early in the Software Development process through the IPT environment. The initial testing of software products will use simulations; however, the introduction of actual hardware products will take place as early in the software life cycle as possible.

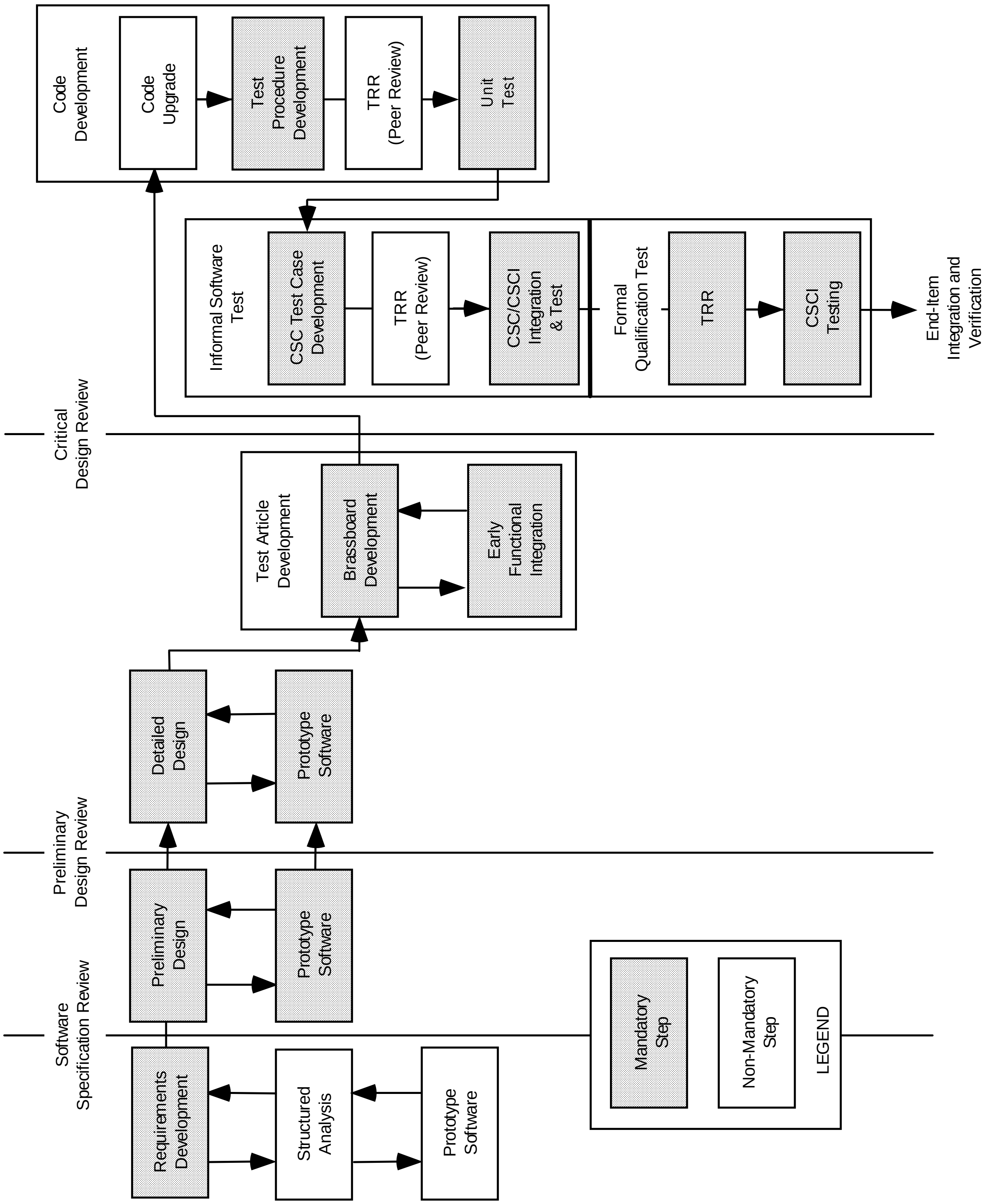


FIGURE 4.2.1.2-1 SOFTWARE DEVELOPMENT PROCESS

TABLE 4.2.1.2-1 DETAILED DEVELOPMENT PROCESS

Phase/Review	Input	Goal	SDF	Products/ Output
Requirements Development Phase	System Architecture, Requirements & Design	Software Requirements. using Structured Analysis	Any trade studies or white papers affecting requirements.	SRSs, ICD Part 1 data identification
Preliminary Design Phase	SW Requirements.	Preliminary Design	Peer review log occurrence, findings and subsequent updates, design decisions & rationale	Preliminary design diagrams (e.g., Buhr), ICD Part 2 message identification
PDR	Preliminary Design	Prime IPT review and acceptance of Prelim. SW Design	Log occurrence, findings and subsequent updates	PDR data delivery products, Updated ICD message content, Formal Review
Detailed Design Phase	PDR Baseline Requirements. update	Detailed Design	Peer review, log occurrence, findings and subsequent updates, PDL	Ada specs, I/F Definitions, SPS, Formal Review
CDR	Detailed Design	Prime IPT Program Review and acceptance of Critical SW Design	Log occurrence, findings and subsequent updates	CDR data delivery products, updated ICD Part 2 message identification, Formal Review
Software Test Article Development Phase	Common Ada Design, SRS, Integrated & informally tested Brassboard code, (Development HW or Simulations as available)	Functionally Correct CSUs, Integrate System SW that resides on multiple platforms, Confirm code meets System Requirements	Log informal Development Status, Log occurrence, date, findings & updates	Compiled, informally tested & functionally verified CSUs, Cross platform system code integration, assurance that code meets System Requirements

TABLE 4.2.1.2-1 DETAILED DEVELOPMENT PROCESS (Concluded)

Phase/Review	Input	Goal	SDF	Products/ Output
Code Upgrade Phase	Brassboard CSUs, Code Review Team Guidelines	Upgrade Code to meet Ada coding standards	Incremental log of code upgrade status	CSUs that meet products standards
CSU TRR (Peer Review)	Upgraded Code & CSU Test Procedures	Assure proper test coverage	Log data under review minutes & actions	Approved test cases with consensus on test case viability
CSU Test Phase	Upgraded Design, Code & approved test procedures	Assure all CSUs are adequately tested	Log date of test, findings & CM code location	Tested CSUs
CSC /CSCI TRR (Peer Review)	Tested CSUs and CSC Test Procedures	Assure all CSCs and CSCIs are adequately tested	Log date under review, minutes & actions	Approved test scenarios with consensus on test case viability
CSC/CSCI Integration and Test Phase	Tested CSUs and approved Test Procedures	Assure all CSC and CSCIs are adequately tested and integrated	Log date of finding & CM code location, As run test cases	Tested CSCs, Integrated CSCIs
NOTE: Expected that Requirements Developers, SW Safety and SW QA be involved through the IPTs as early as possible in the development process. Use of the term software in this table also encompasses the data associated with the software.				

TABLE 4.2.1.2-2 ISS SOFTWARE TYPES VS. MANDATORY DEVELOPMENT STEPS

Phase/Review	Flight SW & Critical Ground SW	MBF SW	GSE/TSE SW	Test SW Including Simulations	SVF SW
Requirements Definition.	X	X	X	X	X
Preliminary Design	X	X	X	X	X
PDR	X	X			X
Detailed Design	X	X	X	X	X
Prototype SW	X				
CDR	X	X			X
Brassboard Development	X	X	X	X	X
Early Functional Integration	X	X	X	X	X
CSU Test	X	X	X	X	X
CSC/CSCI Integration and Test	X	X	X	X	X
X = mandatory development process step					

4.2.1.2.1 SOFTWARE DESIGN METHODOLOGIES AND TOOLS

This section briefly discusses the application of software design methodologies and tools. (In this document the terms techniques and methodologies are synonymous). This document does not levy specific standards for code design methods or tools. The designers have the experience and knowledge to choose the best methods and tools for the software applications they have been given the task to design. Therefore, selection of design methodologies or related CASE tools will be at the discretion of the Tier 1 Subcontractors.

CASE tools are intended to aid in the design and implementation of the software; usually, they do not adequately communicate the higher level design requirements. Although the selection and use of specific methods and tools is not limited, developer selected methods and tools will not affect, constrain, or limit either the presentation of requirements in the SRS documents or the final design in any way.

4.2.1.2.2 REQUIREMENT DEVELOPMENT PHASE

The ISS software requirements will be documented in SRSs. These SRSs will be provided for each CSCI identified in Appendix A of this plan. Use of software functionality and requirements contained in the SSFP FSSRs should be used to the greatest extent possible in the generation of these SRSs. Specific grandfathering of existing FSSRs will be granted by the Prime as outlined in Section 4.2.1.3 of this plan. This section, within the SDP for each Tier 1 Subcontractor, will explicitly identify the SRSs for which they are responsible, any relationship to existing FSSRs, and the detailed process and methodologies to be used in the generation of these specifications.

During this phase, each Tier I Subcontractor may use prototypes and/or models to support requirements analysis.

4.2.1.2.3 PRELIMINARY SOFTWARE DESIGN PHASE

The ISS Software Development methodologies will provide for a Preliminary Software Design resulting in a formal PDR. The formal PDR review will be consistent with Section 3.8 of this plan. Each Tier 1 Subcontractor will perform their risk analysis using Prototype Software to help support early requirements verification and Software Design feasibility.

The following are the minimum requirements to support the Preliminary Software Design phase:

A. Prototype software will be developed for key high program risk and critical functions. The Tier 1 Subcontractor will identify to the C&DH SWI IPT which software is being prototyped;

B. Preliminary bit-level CSCI external interfaces information;

C. The software prototyping efforts that involve feasibility studies, feed back into the output of the Requirements Development phase if a requirement is proved to be unfeasible; and

D. SDFs are complete with PDR-level documentation and available for review at formal PDR.

4.2.1.2.4 DETAILED SOFTWARE DESIGN PHASE

The ISS Software Development methodologies will provide for a Detailed Software Design phase resulting in a formal CDR. The formal CDR review will be consistent with Section 3.8 of this plan. Each Tier 1 Subcontractor will perform their detailed design using prototype software to help support early requirements verification and Software Design feasibility.

The following are the minimum requirements to support the Detailed Software Design phase:

A. Prototype software will be developed for all high program risk and critical functions. The Tier 1 Subcontractor will identify to the C&DH SWI IPT which software is being prototyped;

B. Compiled Ada specifications;

C. Package specifications to describe the interface definitions at a minimum;

D. SDFs are complete with CDR-level documentation and available for review at formal CDR; and

E. Completed bit-level CSCI external interfaces documented in program-level ICDs.

4.2.1.2.5 SOFTWARE TEST ARTICLE DEVELOPMENT PHASE

The ISS Software Development methodologies will provide for demonstration of proper functional requirements rather than via an analysis process. This step is mandatory for all high risk flight software and is called Software Test Article Development phase and will include Brassboard Development and Early Functional Integration. This phase provides feedback to the Detailed Software Design phase to solidify design solutions. All products of the Software Test Article Development phase, including guidelines and completed checklists will be maintained in the SDFs (see Section 4.2.2). It is recommended that informal configuration control processes consistent with Section 7 of this plan be established during this phase. The Tier 1 Subcontractor's SDP will define these internally-controlled processes.

4.2.1.2.5.1 BRASSBOARD DEVELOPMENT

Development of brassboard software supports the "code a little, test a little" method of software building that identifies problems in the requirements at the earliest moment possible in the life cycle. The completion of unit testing will occur during the code development phase. During this phase, a software component will be functionally tested in a "stand-alone" environment, integrated with the simulation software and optionally with MDM executive, C&C processor executive and 1553 communication services. During this phase extensive interaction with software requirements engineers should be taking place. This phase does not require a complete CSCI development.

4.2.1.2.5.2 EARLY FUNCTIONAL INTEGRATION

The Software Test Article Development phase concludes in Early Functional Integration. This early integration of the brassboard code will use realistic scenarios to ensure that the brassboard software provides the functionality to satisfy algorithmic and interface requirements. This activity includes integration of brassboard software with hardware (actual or simulated), firmware, and other CSCIs. This testing will allow risk identification and mitigation before brassboard code is formally brought up to product code standards during the code development phase. A list of findings, significant code changes, and action items resulting from Early Functional Integration will be recorded in the SDF.

4.2.1.2.6 CODE DEVELOPMENT PHASE

The Code Development phase begins by the upgrade from brassboard software to product code by complying with the proper program standards following the completion of Functional Integration testing. It is expected that necessary software test drivers be generated during this phase, and used wherever completed simulations are not available or do not provide for adequate Computer Software Unit (CSU) Testing. Proof of product code and test driver maturity to perform CSU testing will be assured before proceeding to CSU testing. The recommended, but optional, approach of this assurance is through conduct of an informal TRR (i.e., Peer Review). The Code Development phase is considered complete at the end of CSU testing. All products of the Code Development phase, including guidelines and completed checklists will be maintained in the SDFs (see Section 4.2.2).

4.2.1.2.7 CSC/CSCI INTEGRATION AND TESTING PHASE

The CSC/CSCI Integration and Testing phase will begin with review of the output from the Code Development phase to assure all actions have been completed together with proof of product code and test case maturity to perform CSC/CSCI testing by a TRR (i.e., Peer Review) or equivalent process. During this phase, the Tier 1 Subcontractor will ensure that the algorithms and logic employed by each CSC is correct and that the CSCs integrate into a CSCI which satisfies its specified requirements. The record of test results of all CSC/CSCI integration and testing will be maintained in the SDFs.

The Tier 1 Subcontractor will make all necessary revisions to the design documentation and code, perform all necessary retesting, and update the SDFs of all CSUs, CSCs and CSCIs that undergo design or coding changes based on the results of all testing performed.

Formal Qualification Test will begin with a TRR to review the output of the CSC and CSC/CSCI integration and testing together with proof of product code and test case maturity to perform FQT. SCM (per Section 7) is required to maintain baseline control of the CSCI source and object code being tested through the FQT phase and support to all further ISS integration milestones. Details of FQT are provided in Section 5. Product compliance with Software Safety and Software Quality is assured during this phase.

4.2.1.3 SPACE STATION FREEDOM DERIVED PRODUCTS

Major portions of the SSFP software products are applicable to the ISS program. Those products determined to have completed major software program milestones (e.g., CDR) are eligible for grandfathering for the ISS program. This grandfathering approval will be granted by the Prime, on a case-by-case basis. To obtain this approval, the Tier 1 Subcontractor will explicitly identify the system-level CSC/CSCI requiring less than 20% modification from SSFP, as well as proof of milestone completion. Relief from products and processes identified below, will be granted after the C&DH SWI IPT has evaluated the total impact to the CSC/CSCI to determine the overall CSC/CSCI grandfathering.

A. SRSs will be written for all ISS products per Table 3-1. SSFP FSSRs can be referenced from the SRSs, to provide the software requirements where functionality has not changed from SSFP to ISS. The SSFP FSSRs will then be packaged together with the ISS SRSs and delivered to the Prime as a Type 2 submittal.

B. PDR, Detailed Software Design and CDR milestone duplication is not required, based on proof of milestone completion on SSFP. All proof of these processes and milestone compliance will be maintained in the SDFs.

C. Brassboard Development, Early Functional Integration, CSU TRR and CSU Test are not required based on proof of milestone completion on SSFP, together with regression testing of all CSUs with the ISS C&C and services products. All proof of these processes and milestone compliance will be maintained in the SDFs.

D. CSU and CSC to CSCI Integration testing is not required, based on proof of milestone completion on SSFP, together with regression testing of all CSCs and CSCIs with the ISS

C&C and service products. All proof of these processes and milestone compliance will be maintained in the SDFs.

E. Relief from Formal Test processes and milestones will not be granted for the ISS program.

A PDR/CDR delta-review will be held to formally accept the (less than 20%) modifications to the SSFP product to support the ISS program. The exact methodology for this review will be defined in the Tier 1 Subcontractor SDP as approved by the Prime.

4.2.2 SOFTWARE DEVELOPMENT FOLDERS

SDFs are the repository for all information relative to the development of a software product. The folders are used during the development process to collect and capture all data relevant to the product and to communicate the development status with the developer's peers and management. The folders are also used to support the maintenance, design trades, and developmental testing. The SDFs may be in contractor format, but must contain the contents specified in the SSPS. The SDFs replace the SDD as the location of the Section 4.0 Detailed Design information, including Program Design Language (PDL) or other detailed design diagrams. The SDFs are reviewable by the Prime, NASA or IV&V at any time.

4.2.3 DESIGN STANDARDS

To promote consistency in implementation of requirements, the flight software design standards and guidelines, defined below and in the SSPS will be followed by all Tier 1 Subcontractors to the extent possible as they apply to their specific subsystems. The Tier 1 Subcontractors may be granted an exception to some design standards due to the state of the design and development of their software under the SSFP program, as long as they comply with all interface control requirements without jeopardizing the integration of all subsystems in the ISS program. Any exceptions to these design standards must be identified in the Tier 1 Subcontractor's SDP and approved by the C&DH SWI IPT.

The design guidelines in this section should be used as a complement to the Prime SSPS.

The specific design standards include:

A. Naming Standards;

B. Interface Standards; and

C. MDM Adaptation Data Overlay (ADO) Standards.

4.2.3.1 NAMING STANDARDS

Naming standards are intended to promote a common usage of terms, understanding of concepts, means of identification and communications among the software community and with all other program disciplines. The nomenclature, data and signal naming standards will be established and documented in the SSPS.

4.2.3.2 INTERFACE STANDARDS

The software interfaces within an Ada program will be expressed in the form of Ada specifications conforming to the coding standards defined in the SSPS. The interface standards between CSCIs will be in conformance with the information supplied in the ICD specific to each subsystem.

4.2.3.3 MDM ADAPTATION DATA OVERLAY STANDARDS

The Adaptation Data Overlay Standard defines the format of USOS MDM resident data tables that may be modified from the USGS using Pre-Positioned Load (PPL) files (See Software Interface Control Document, Part 1, United States On-Orbit Segment to United States Ground Segment, Command And Telemetry (ICD Part 1), SSP 41154).

The SSPS contains additional information on USOS ADOs and Pre-Positioned Load Files.

4.2.3.3.1 TERMINOLOGY

Adaptation Data is contained within each MDM's Flight Software (FSW) Application load image. This data is "mostly" constant. To FSW Applications it is Read-Only information. To Ground Controllers it is operational data that may have to be modified at run-time to "tune" the flight system or adjust for hardware configuration changes. Command routing tables, time drift compensation parameters and high/low limit values are examples of Adaptation Data.

(Note: the I/O Configuration Table used by the MDM Boot Software to check an MDM's I/O card complement is Adaptation Data that is not part of the MDM's Application load image.)

Adaptation Data tables may be modified on-orbit between releases of an MDM's software load.

Adaptation Data Overlays are a subset of Adaptation Data Tables. Each ADO includes a control header that contains version and checksum information. The ADO version ID identifies the specific instance of data in the table. The checksum information verifies table contents and allows a new data set to be loaded without revising the total checksum for the larger block that includes the ADO.

Pre-Positioned Loads are files maintained on the ground that contain data to revise the default data contained in the ADOs. PPLs have a header of their own. Each PPL header contains the information needed to load it on-orbit (See ICD Part 1).

The difference between the terms ADO and PPL is the ADO is a design construct for the subset of adaptation data tables that are likely to be changed and need special consideration, where PPLs are load files containing data for the overlay of default data in the ADO.

4.2.3.3.2 ADO AND PPL SPECIFICATIONS

The following specifications apply to ADOs and PPLs. Detailed formats for defining ADO data structures and creating PPLs are specified in the SSPS.

A. Adaptation Data Overlays are USOS MDM FSW Adaptation Data tables designed to be modified on-orbit by Data Load commands from the ground.

B. Ground control uses PPLs to upload ADO tables to FSW load images in 1) DRAM and 2) EEPROM or to the CCS MSD.

C. Each ADO is a contiguous block of data.

D. The first two words of an ADO contain a version ID and checksum wild card. The version ID identifies what revision of the ADO is operational. The checksum wild card verifies ADO contents.

E. The ADO Version ID is assigned a PUI that may be retrieved for operator display. Note: This is an ICD requirement which will continue to use the term “PPL Version ID” (See Software Interface Control Document, Part 1, Station Management And Control to International Space Station, Book 2, General Software Interface Requirements, SSP 41175-2).

F. Each released version of MDM FSW contains at least one version of each of its ADOs. Any additional ADO versions required for assembly operations that occur prior to the next software release are delivered as PPLs.

G. ADOs are developed, tested and loaded in their entirety. They are not patched “piecemeal”.

H. An ADO is not delivered as a PPL until the default values in the ADO need to be updated. When a second version of an ADO is created, as a PPL, the PPL for the first version is also created. This allows Ground Control to recover to a known baseline if errors occur using the second PPL version. PPL’s for subsequent versions of ADO modifications are created as needed to meet operational needs.

I. The adaptation data overlay design description documentation is contained in the standard design documentation of each CSCI. See the SSPS for detailed guidelines.

4.2.3.3.3 PPL FLIGHT AND MATURITY LEVELS

Rev B

PPLs will be assigned Maturity levels according to their current place in the development and certificaiton life-cycle. Table 4.2.3.3.3-1 outlines the maturity levels for PPLs on the ISS program. A PPL must have a maturity level of Flight prior to being incorporated into a Flight IFL. All other maturity levels may be released to the ISS program in non-flight IFLs.

Rev B

TABLE 4.2.3.3.3-1 PPL MATURITY LEVEL DEFINITIONS

Maturity Level	Criteria
Engineering	Implies the PPL instance has been released, AS IS and with out guarantee, to support a program need. The developer has not yet begun to integrate the PPL into the associated CSCI.
Integration	Implies the PPL instance has been released, AS IS and with out guarantee, to support a program need. The developer is performing internal integration testing/analyses work.
FQT	Implies the PPL has completed the developer required qualification steps specified in the Requirements Data File (RDF) for the PPL instance. The qualification requirements documented in the RDF may refer to any of the formal qualification methods: inspection, analysis, demonstration, or test. If Stage Verification testing is not required by the RDF, then FQT Maturity of a PPL instance will move directly to Qualification Maturity.
Qualification	Implies the PPL has completed ALL internal integration testing/analyses qualification activities required to prove compliance with the RDF content. Qualification Maturity is required before a PPL instance can be included into a Flight IFL.
Flight	Implies COMPLETION of all testing/analyses activities for qualification and all subsystem performance analyses required to show use of the PPL instance will result in acceptable on-orbit performance of the ISS vehicle. The Flight Maturity level may be reached immediately after Qualification Maturity if and only if no further analysis is required to certify the PPL instance for on-orbit use.

4.2.4 CODING STANDARDS

The primary languages to be used for developing all U.S. Space Station software will be Ada and C. The use of Ada will adhere to the software coding standards contained in the SSPS. This software includes onboard flight software, ground support software, support and development software (e.g., SEE), simulation software, test software, and training support software. The SSPS specifies the coding standards which are to be applied to all software developed in Ada and additional restrictions that apply to the on-board flight software and safety critical ground support software developed in Ada.

Exceptions to the use of Ada or C are anticipated for flight software such as COTS, firmware controllers which operate in processors non-conducive to programming in Ada or C, time critical functions where Ada or C will not meet timing constraints and low-level bit manipulation functions. The flight software master CSCI list in Appendix A identifies the languages used for each flight CSCI.

Use of languages other than Ada or C for flight software will be requested by each Tier 1 Subcontractor and authorized by the SWI IPT. A request will be made for each CSCI and include the following types of information:

A Language requested, processor type and function or CSC/CSU where the language will be used;

B. Risks associated with the use of the language;

C. Maintenance requirements, including compilers, development/tool environment required to maintain the software in this language;

C. Operating and integration requirements, including any special software required to operate or integrate this software due to the language; and

D. Identification of the coding standards used for this software.

The request will be submitted to the C&DH SWI IPT and represented at the designated IPT meeting by the requesting Tier 1 Subcontractor. If all relevant information is correct but the SWI IPT team leaders and requesting Tier 1 Subcontractor cannot reach agreement, the request is elevated to the C&DH Management Team for resolution. Approved requests will be maintained by the Prime in the SCM Library System. The process flow and request forms for this activity will be maintained by the C&DH SWI IPT as working desk procedures.

4.3 NON-DEVELOPMENTAL SOFTWARE

Non-Developmental Software (NDS) includes reusable software, commercially-available software and Government-furnished software. The use of non-developmental software is allowed on the ISS program with C&DH SWI IPT approval. The use of GFE software must be specified in SDP paragraph 3.1.2. Each of these three areas of NDS are addressed in the following subsections. Detailed descriptions of NDS items which are to be used for the ISS are identified in the lower level SDPs, as appropriate.

Compliance to non-developmental software requirements will be evaluated utilizing the following criteria:

A. Commercial, government, internal manuals or specifications, demonstrated results, test reports, or other performance data exist, prior to its incorporation, evidencing that the software meets requirements;

B. Software is placed under configuration control prior to its incorporation. This ensures that upgrades or changes that alter software or system operation are properly qualified. In-house modifications to non-developmental software will be treated as new development; and

C. Data rights and provisions are, by extension, the same as those required by the contract.

4.3.1 REUSABLE SOFTWARE

Tier 1 Subcontractors should aggressively pursue software reuse. Software will be considered for reuse under the following circumstances:

A. The software to be reused meets ISS functional requirements allocated to it;

B. All data items associated with the software meet format and content standards required for ISS software; and

C. Reusable software provides a low-risk, low-cost approach to meeting ISS software requirements. However, software must also meet all process requirements as specified in this SDP.

4.3.2 COTS SOFTWARE

As with reusable software, commercially-available software is a low-risk, low-cost alternative to developing new software and is encouraged for the Space Station. COTS and Modified COTS (MOTS) software may be included in deliverable software and must follow the documentation and development approach defined in this section.

Modified portions of COTS must comply with the same development and documentation standards and procedures as newly developed software. Supplier documentation can be included in ISS documents by reference, but must be augmented if the intent of the ISS DID requirements are not otherwise met. An SPS is not required if no new code is added or changed, and the source code listings are supplied.

Test plans and test descriptions must be produced but will not require as exhaustive a testing regimen as newly developed code. Interfaces and hardware-peculiar aspects must be fully tested. Software Test Reports must be produced for all tests performed. The verification and validation approach for COTS software is scoped according to the origin and assessed risk of the software. COTS software is not required to be informally tested and is allowed to contain unused logic. FQT will test only the function, performance and compatibility at the product level unless there is reason to believe that the COTS logic is causing logic errors. A VDD must be supplied but the requirement can be met through reference to supplier documentation. Executable code must also be included. A User Guide from the supplier is acceptable if the DID content requirements are satisfied. Configuration management of COTS software will be handled in the same fashion as newly developed software.

The following criteria will be used for specific types of COTS software:

A. Embedded COTS software should be documented with the CSCI to which it belongs, even though some elements of the CSCI are not COTS. COTS software should be clearly identified in the SPS with a brief description included, besides the documentation included by reference;

B. Pre-Existing COTS and MOTS software and firmware will be delivered with the documentation and products as they were available before novation, adding only those which can be obtained at no additional cost to ISS; and

C. Ground-based COTS support software used for general purposes (e.g. compilers, test or debug tools) will be delivered with only the commercially available documentation, the executable code and a VDD. PIDS requirements and general functionality will be verified during system level testing using the Acceptance Test Procedure (ATP).

Boeing will negotiate licensing agreements for COTS software tools required team-wide. These agreements will have provisions for transfer of licenses to the Government to assure coverage of total system life cycle.

4.3.3 GOVERNMENT FURNISHED SOFTWARE

The Government Furnished Software used on the ISS program is identified in Section 3.1.2 of this plan and the Prime GFEL included in the SOW.

4.4 NON-FLIGHT SOFTWARE

Non-flight software comprises software that is used to support ground activities including: ISS flight software design, development, integration and verification; flight article and end item design, development, qualification and acceptance; on-orbit stage configuration integration and verification; and, launch package integration. This software is categorized into the following groupings:

A. Test software, including simulations;

B. SVF software;

C. GSE/TSE software; and

D. Ground software, including MBF software.

4.4.1 TEST SOFTWARE AND SIMULATIONS

Test software and simulations will provide the environment for the integration and verification of the ISS flight software and avionics from software development through integration. This will involve the test environment with end item and segment simulations; environmental simulations which represent the ISS on-orbit environment and dynamics; and, sensor/effector simulations. Some simulations will be reused in verification activities at multiple facilities (i.e., end item, Stage, and Launch Package).

The four different types of test software are vertical simulations, horizontal simulations, test software and test configured flight software. Vertical simulations are simulations specifically developed to support CSCI testing. Horizontal simulations may be either vertical simulations enhanced to meet horizontal simulation requirements or newly designed simulations required for horizontal testing and not available from vertical sources. Test software (i.e., stubs, drivers, etc.) is software specifically written to meet a specific testing goal. Test configured flight software is flight software built to a test specification. This type of software will be used when flight software is available from CM for modification or augmentation to meet specific test needs.

4.4.1.1 TEST SOFTWARE AND SIMULATION LIFE-CYCLE

The test software and simulations used in the integration and verification process of flight software and avionics will follow the same general development life cycle and configuration management controls as that established for the flight software defined herein, but may be tailored by each subcontractor to be more cost effective by use of existing plans and deliverables definition. Subcontractor test software and simulation life cycle reviews, documentation, and verification planning and processes, including appropriate tailoring, will be included in the Tier 1 Subcontractor's SDP and be subject to the approval of the C&DH SWI IPT.

4.4.1.2 SIMULATION REUSE

To minimize verification costs and maximize simulation reliability, the ISS program will reuse simulations from CSCI and end item verification test environments in the SVF. The Prime will coordinate upper level verification simulation reuse requirements with the Tier 1 Subcontractors for implementation into the Tier 1 Subcontractors simulation requirements. Any Prime requested Tier 1 Subcontractor simulation requirement changes for SVF use will be via normal engineering requirement change instruments. The Prime is responsible for porting of the simulations to the SVF simulation environment. The Tier 1 Subcontractors will support the integration, verification, and certification activities, as well as, maintenance and technical support for the simulations to be reused. The identification of these simulations and the reuse development plan is defined in Appendix A of S684-10140, the SVF PIDS.

4.4.1.3 SIMULATION CERTIFICATION

Before use of the flight simulation software in the FQT of the flight software, a set of informal tests will be conducted to assure the simulation software accurately provides the stimuli and interface data defined by the relevant Orbital Replacement Unit (ORU) specifications and ICDs. This testing will be accomplished with the concurrence/approval of the design team responsible for the item being simulated. Formal verification for the simulation software will not be conducted, however the simulation software will be classified as "certified by use" following completion of the informal testing and successful use of the simulation in the flight software FQT.

4.4.2 SVF SOFTWARE

The SVF software includes both real-time and non-real-time test environment support CSCIs. The real-time test software will follow the life cycle as described for the test and simulation software with any additional tailoring included in the facility development plan. The non-real-time software will support pre-test, configuration management, facility unique operations, and flight software verification tracking. COTS software must meet the provisions as defined in Section 4.3.

4.4.3 GSE/TSE SOFTWARE

GSE/TSE, as defined in the Prime Integrated Logistics Support Plan, D684-10041-1, that interfaces with the flight hardware must be verified to the same levels as the flight software, but is not usually as functionally complex as other types of software. GSE/TSE also contains a large percentage of COTS software and small percentages of critical new development software. The development life cycle for this type of software does not require the same rigor of design and development documentation as other non-flight software (e.g., software test plan information may be included within the hardware acceptance test plan). Verification of GSE/TSE software will be rigorous to ensure proper interfacing with flight hardware. The Tier 1 Subcontractor's SDP will identify GSE/TSE that includes complex, mission critical modified COTS or new development software, and define the development life cycle, the deliverable documentation and data rights and provisions (see Section 4.3). GSE/TSE using unmodified COTS or small percentages of developed software will be identified as part of the GSE/TSE development process and will be required to meet the data rights and provision requirements of Section 4.3.

4.4.4 GROUND SOFTWARE

Ground software supports the on-orbit mission and includes crew and payload training, CCC operations, analytical tools, models and the MBF. Training and operations software is not the responsibility of the Prime. Ground software, as identified in Appendix A of this document or the Tier 1 Subcontractor's SDP, may follow a tailored software development life cycle and include compressed life cycle phases and documentation and an informal review process.

4.5 MDM COMMON SOFTWARE

MDM Common Software includes MDM Boot and Diagnostic software and MDM Utilities software developed by the MDM provider and MDM Utilities Extension software coordinated and developed by the Computational Systems Implementation Team, a sub-group of the C&DH SWI IPT, and now being supplemented and enhanced by the C&C Software IPT. The MDM Boot and Diagnostic software is a common firmware Controller CSCI in each MDM which performs the Boot, Power On Self Test, Data Upload and Data Download functions. The MDM Utilities software performs MDM hardware access and health monitoring functions for all MDMs and file management services for the internal mass storage device of the Enhanced MDMs. The MDM Utilities Extension software is a common set of software which provides the cyclic scheduler, data transfer and the command interfaces mechanism, performs the 1553 communication interface and provides a list based sensor/effector Input/Output (I/O) mechanism.

4.5.1 MDM COMMON SOFTWARE LIFE-CYCLE

The scope of MDM Common Software has been reduced from the SSFP baseline on the ISS program. Rapid modifications to these services will begin following ISS SRR with the Brassboard Development phase of the software development process. Modifications will be made on a function by function basis with informal deliveries made to Tier 1 Subcontractors. The order of MDM Common Software modifications will be established in conjunction with Tier 1 Subcontractors and the Prime. Informal delivery of MDM Common Software will be accompanied by release notes that will include instructions for configuration and use.

4.5.2 MDM COMMON SOFTWARE TESTING/ACCEPTANCE

Final delivery of MDM Common Software will follow extensive Tier 1 Subcontractor use. Extensive CSC testing will result from Tier 1 Subcontractor integration and testing of ISS CSCIs. The MDM Boot and Diagnostics software and MDM Utilities software will be formally tested by the provider prior to final delivery to the Tier 1 Subcontractors. The MDM Utilities Extension software will be provided to the developer's as a reuse product and will not be formally tested prior to delivery. Each user will formally test the pieces of the MDM Utilities Extension software they use during the FQT of the CSCI in which the pieces were incorporated.

5 FORMAL QUALIFICATION TESTING

FQT is a process that allows the Prime to determine whether a software configuration item complies with the allocated requirements for that item. FQT will be conducted in accordance with guidelines provided by DOD-STD-2167A as tailored in this plan.

Tier 1 Subcontractors will develop a STP for conducting the formal qualification testing activities required by DOD-STD-2167A and this plan. Following C&DH SWI IPT approval of the STP, Tier 1 Subcontractors will conduct testing in accordance with the STP. With the exception of scheduling information, updates to the STP will be subject to C&DH SWI IPT approval. Schedule control will be maintained by program control activities as defined in Section 3.2 and not in the SDP or STP.

FQT will be conducted on flight and ground software. Verification will be at a level defined by the Tier 1 Subcontractor and subject to C&DH SWI IPT approval.

Figure 3-2 illustrates a software and formal qualification testing life-cycle. The software development activities specified by DOD-STD-2167A are represented at the top of the diagram from left to right, flowing downward toward “CSCI Testing”. CSCI test preparation activities are illustrated beginning with test planning flowing upward, left to right, toward “CSCI testing”. Software development FQT activities conclude with CSCI testing. Software CSCI development and software test preparation activities are conducted separately for all software CSCIs under development. CSCI testing is the responsibility of Tier 1 Subcontractors developing flight software, firmware and support software.

End Item Integration and Testing represents instances where CSCIs are integrated with other CSCIs and Hardware Configuration Items (HWCIs) associated with a specific end item and is the responsibility of the Tier 1 Subcontractor. Stage I&V represents horizontal integration and verification activities at the stage level and is the responsibility of the Prime. Both of these tests are outside the province of this SDP, but are discussed briefly in this section to aid in understanding their relationship to FQT.

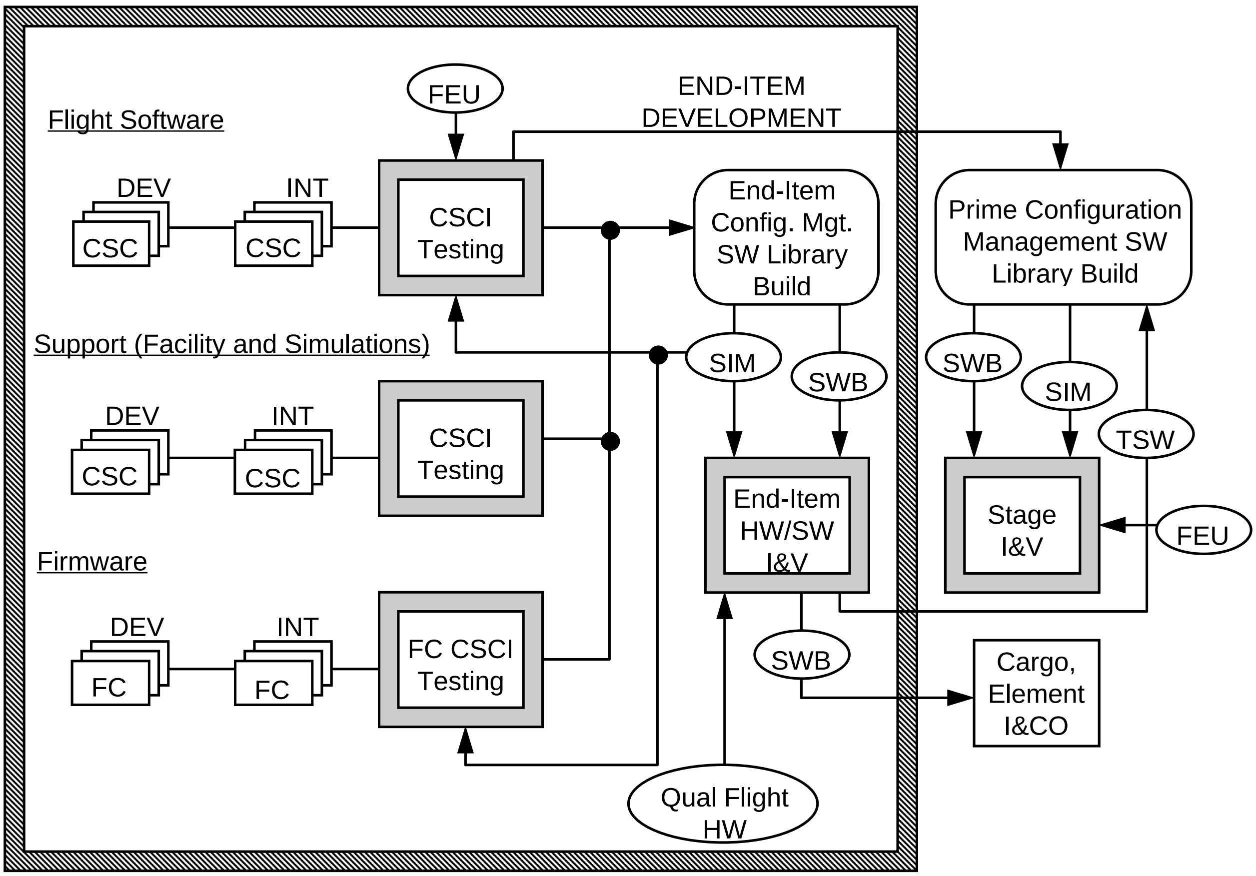
Figure 5-1, ISS Software Verification Approach, provides an illustrated view of the ISS verification approach that includes development and test activities across the program. The FQT scope ends with CSCI Testing. However, should requirements not be verifiable during CSCI testing they may be allocated upward to End Item HW/SW I&V, and Stage I&V. These areas where CSCI requirements may be verified are indicated by shaded boxes. For clarity, hardware activities that influence or are influenced by FQT are included. Further details on the FQT approach and philosophy are supplied in Section 5.2.

<u>Delta-FQT:</u>

Post-CSCI FQT fixes to reported problems may be tested without a full FQT. To accomplish this, a Delta-FQT will be used. A Delta-FQT will be conducted as a nominal FQT except its tests are specifically targeted. A Delta-FQT will include tests for specific problems fixed along with regression tests as outlined in the TRR. Successful completion of the Delta-FQT process is the same as for nominal FQT but limited per the TRR.

Incremental FQT:

An IFQT will be used in support of ITRRs. The IFQT process provides for testing portions of a CSCI as outlined in the corresponding ITRR. The IFQT will be conducted as a nominal FQT except its tests are limited to a portion of a CSCI. An IFQT will include tests for the part of the CSCI being tested, tests for specific problems fixed or functionality added in the ITRR/IFQT process along with regression tests as outlined in the ITRR. Each IFQT will add to the number of tests completed on a CSCI with the final IFQT running the last of the tests. Successful completion of the IFQT process is the same as for nominal FQT and should look as if the IFQT process was a nominal FQT.



Legend:

FEU	Functionally Equivalent Units	TSWB	Test Configured Software Build
FSW	Flight Software	TSW	Tested Software
FW	Firmware	SWB	Flight Software Build
DEV	Development	INT	Integration
		I&CO	Integration and Check out

FIGURE 5-1 ISS SOFTWARE VERIFICATION APPROACH

5.1 ORGANIZATION AND RESOURCES

This subparagraph provides guidance relating to the organization of Formal Qualification Test teams.

5.1.1 ORGANIZATIONAL STRUCTURE - FORMAL QUALIFICATION TESTING

Prime and Tier 1 Subcontractor test organizations will be structured consistent with current AIT/IPT structures. Test teams should be organized separate from software development product teams. Test team structures will be defined in Prime and Tier 1 Subcontractor test plans.

5.1.2 PERSONNEL - FORMAL QUALIFICATION TESTING

Personnel selected for test teams will be at the discretion of the Tier 1 Subcontractor test teams conducting the test. Details of the personnel included on the test teams will be included in the Tier 1 Subcontractor SDPs.

5.2 TEST APPROACH/PHILOSOPHY

The ISS approach to FQT ends at CSCI Testing. However, if limitations of the test environment at the CSCI Testing level preclude the verification of all requirements then it is allowable to allocate those requirements upward to end item and/or Stage Integration levels (reference Figure 5-1). The following paragraphs expand the approach and provide a philosophy that supports FQT decisions and activities.

5.2.1 APPROACH

FQT begins with informally verified software CSCIs that have been pre-qualified as ready for integration with their respective processors. The CSCI is then verified as a stand-alone load. Successful load verification demonstrates that the CSCI, to the extent possible, is capable of performing as specified. Figure 3-2 illustrates the scope of FQT and identifies it as “CSCI Testing”. Afterward, the CSCI, now a verified load at the end item level, is ready for End Item HW/SW I&V. Successful HW/SW I&V indicates the end item is now ready for Stage I&V. Details beyond the scope of this plan may be found in the PMI&VP.

5.2.1.1 CSCI TESTING

CSCI Testing is the first formal verification activity following informal integration of CSCI components, or functional CSCs, into a single processor load. The processor load should be integrated informally prior to formal CSCI verification to ensure it will compile, load, and execute in a development environment in accordance with design specifications. Table 5.2.1.1-1, CSCI Testing, expands on the concept of CSCI Testing. Subparagraphs 5.2.1.1.1 and 5.2.1.1.2 address the applicability of flight and support software to Table 5.2.1.1-1.

TABLE 5.2.1.1-1 CSCI TESTING

Type:	CSCI Testing involves the formal qualification testing of the CSCI and interfaces to other CSCI and HWCI components. Testing with HWCI components satisfies the requirements for HW/SW integration at the box level.
Requirements to be Verified:	Requirements verified by this formal activity are those allocated to FQT, at this level, by the Verification Requirements Matrix in the SRS and ICD.
Test Configuration:	The test configuration for this activity is at the Tier 1 Subcontractor's discretion and subject to approval by the Prime. The final configuration is subject to approval by the Prime.
Responsibility:	The responsibility of CSCI testing belongs to the Tier 1 Subcontractor responsible for the development of the CSCI under test.
Location:	The selected location for CSCI testing is at the Tier 1 Subcontractor's discretion and subject to approval by the Prime.
Verification Documentation:	The products of formal verification at this level will be test plans, test descriptions, and test reports prepared in accordance with this plan. Verification products are required to be visible on the Tier 1 Subcontractor document tree provided within respective subcontract Verification Plans. Products targeted for inclusion in Verification Plans are subject to approval by the Prime.
Next Test Activity:	CSCI testing is followed by End Item HW/SW I&V (Reference subparagraph 5.2.1.2 and Table 5.2.1.2-1) and Stage I&V (Reference subparagraph 5.2.1.3 and Table 5.2.1.3-1) which may occur concurrently.

5.2.1.1.1 FLIGHT SOFTWARE

Table 5.2.1.1-1 applies to all flight software CSCIs and includes firmware developed for use with FCs, MDMs, Sensors, Effectors, and other ISS defined flight ORUs.

5.2.1.1.2 SUPPORT SOFTWARE

There are four categories of software to be considered as support software. Those categories are SVF, Test, GSE/TSE and Ground software. CSCI testing for each category is addressed separately in the subparagraphs that follow.

A. SVF

SVF software includes all software required to support, control, or run the SVF. Software CSCIs developed specifically for the facility will follow the same life-cycle as flight software and will adhere to Table 5.2.1.1-1. Requirements not verifiable at the CSCI level may be allocated upward to end item verification.

B. Test

The four different types of test software are vertical simulations, horizontal simulations, test software and test configured flight software.

1. Vertical simulations are tested in their intended environment to ensure they meet the intended use and represent the item simulated. Vertical simulations should be archived at a CM level commensurate to their use and made available for inspection should subsequent examination be required. Vertical simulations may follow a modified life-cycle as specified in the Tier 1 Subcontractor SDPs and do not require adherence to Table 5.2.1.1-1.

2. Horizontal simulations may follow a modified developmental life-cycle as specified in Tier 1 Subcontractor SDPs for simulations.

3. Test software is not deliverable nor does it require adherence to the full software life-cycle or Table 5.2.1.1-1. If it is used during requirement verification, it must be placed under CM for future use should an anomaly point back to the verification actions as being suspect. The justification and design information for test software should be placed in appropriate SDFs.

4. Test configured flight software does not require adherence to Table 5.2.1.1-1, but does require traceability to STDs and procedures. If test configured flight software is used during requirement verification, then the representative build should be maintained under CM for future use should an anomaly point back to the verification actions as being suspect.

C. GSE/TSE Software

GSE and TSE software will be accepted and/or qualified according to the size and amount of new software development, its use, criticality, and longevity when supporting flight article verification. GSE/TSE software does not require full adherence to the software life-cycle or Table 5.2.1.1-1.

D. Ground Software

Ground software does not require full adherence to the software life-cycle or Table 5.2.1.1-1. Ground software will require qualification with the STP, STD, and a STR with as-run test procedures.

5.2.1.2 END ITEM HW/SW INTEGRATION AND VERIFICATION

End Item HW/SW I&V assures successful integration of CSCIs associated with each end item and is discussed in this SDP only to aid in the understanding of its relationship to software testing and FQT. Table 5.2.1.2-1, End Item HW/SW I&V, expands on the concept of End Item HW/SW I&V as it applies to this program.

TABLE 5.2.1.2-1 END ITEM HW/SW INTEGRATION AND VERIFICATION

Type:	End Item HW/SW I&V supports a single end item. Verification activities include CSCI interface verification to other CSCIs, HWCIs, and end items.
Requirements to be Verified:	Requirements to be verified include requirements allocated to this level by the Verification Matrix at the B1 level for the end item and software CSCI requirements allocated to this level from CSCI testing.
Test Configuration:	The test configuration for this activity is at the Tier 1 Subcontractor's discretion and subject to approval by the Prime.
Responsibility:	End Item HW/SW I&V is the responsibility of the Tier 1 Subcontractor responsible for the end item.
Location:	The selected location for End Item HW/SW I&V will be at the Tier 1 Subcontractor's discretion and subject to approval by the Prime.
Verification Documentation:	The products of formal verification at this level will be test plans, test descriptions, and test reports. Verification products are required to be visible on the Tier 1 Subcontractor document tree provided within respective subcontract Verification Plans. Products targeted for inclusion in Verification Plans are subject to approval by the Prime.
Next Test Activity:	Stage I&V. End item flight article integration may occur concurrently. Reference Paragraph 5.2.1.3 and Table 5.2.1.3-1.

5.2.1.3 STAGE INTEGRATION AND VERIFICATION

Stage I&V is the last series of verification activities prior to On-Orbit Assembly and Operations and is discussed in this SDP only to aid in the understanding of its relationship to software testing and FQT. Table 5.2.1.3-1, Stage I&V, further expands on expected verification activities.

TABLE 5.2.1.3-1 STAGE INTEGRATION AND VERIFICATION

Type:	Stage I&V involves the verification of requirements and functions allocated to the specific stage under test.
Requirements to be Verified:	Requirements verified are those allocated to a specific stage by the USOS Specification, operational requirements, requirements allocated upward from lower level specifications, and requirements that illustrate that the performance meets user expectations.
Test Configuration:	The test configuration is the responsibility of the Prime and will consist of an SVF capable of providing subsystem simulations, sensor effector, and firmware controller simulations and simulators in sufficient quantities to provide a simulated multi-stage environment under control of an automated host test computer system.
Responsibility:	Stage I&V is the responsibility of the Prime. At the discretion of the Prime, participation of end item Tier 1 Subcontractors may be required.
Location:	JSC.
Verification Documentation:	Verification products are required to be visible on the document tree provided within the Program Integration & Verification Implementation Plan.
Next Test Activity:	KSC assembly and checkout.

5.2.2 FQT PHILOSOPHY

The philosophy of FQT supports a process that allows the Prime to determine whether software configuration items comply with the allocated requirements for that item. FQT will begin with CSCI testing and through the allocation of specific requirements upward, may continue through End Item HW/SW I&V and Stage I&V.

CSCI requirements are found in SRSs and ICDs. Because SRS requirements are allocated from higher level specifications, it is possible that some requirements such as performance, timing, or external interface requirements cannot be totally verified in a pure software environment. In these instances it is permissible to allocate them upward to end item testing where adequate hardware/software integration testing is supported. Additionally, any CSCI requirements that the developer desires to be verified through the accomplishment of informal tests must be pre-declared in the Tier 1 Subcontractor's STPs.

The definition of CSCI verification requirements (SRS paragraph 4.2) are used to scope the verification effort by stating the amount and nature of the stimuli to be applied to the software in verifying compliance to a particular SRS requirement. Writing verification requirements provides a straight forward way to answer the question: "When is testing complete?". Testing is complete when every SRS requirement has been exercised to the extent specified in the verification requirement.

Flight Software CSCIs residing in their target processors often will not be physically co-located with all the end item hardware which they functionally control. This causes software planning to take into account the hardware/software integration of CSCI interfaces to sensors, effectors, and firmware controllers remotely located on these end items. Verification will require exercising the end item with hardware and software interfaces provided by simulations (or simulators). Flight software running in a target processor will be under control of a software test environment specifically configured for the end item under test. It is also possible that multiple CSCIs will have interfaces to an end item. In this configuration, multiple CSCIs will be running in their respective target processors under control of the software test environment.

Fully tested end items integrate with other on-orbit end items to become Stages. Each launch creates a new Stage. Therefore the new pieces that will be added to the existing Stage require verification under an environment that simulates the Stage it will join. Requirements verified at this level are those allocated to Stage I&V or validate the intended use of the Stage once it is on-orbit. Software verification of Stage requirements will take place in the SVF.

Following successful end item testing, the integrated hardware and software is packaged and shipped to KSC for launch. The software will concurrently be resident in the MBF, where a comparison will be made to ensure the same load exists at both the MBF and KSC. The end items will be unpackaged at KSC and put through an unpacking check to ensure no shipping damage resulted from the transport. KSC will repackage the components for launch.

Facilities and horizontal simulation software require integration and verification activities in conjunction with the qualification of the SVF. The target processors and interfaces for this software is the SVF itself. A consequence is that test planning is highly dependent on SVF availability for CSCI FQT activities. For support software such as facility and horizontal simulation CSCIs, requirements may be allocated upward to the SVF which is their target end item.

During all FQT activities, the software verification requirements traceability process will be implemented as described in the PMI&VP.

5.3 TEST PLANNING ASSUMPTIONS AND CONSTRAINTS

The assumptions and constraints contained in the following subparagraphs should be considered during test planning.

5.3.1 ASSUMPTIONS AND CONSTRAINTS

The following assumptions should be considered during test planning:

A. Test facilities are fully certified and current;

B. Versions of target software are available from SCM as required;

C. Sufficient certified simulations are available to provide required test fidelity; and

D. Ready access to current requirements data base.

5.3.2 CONSTRAINTS

Test planning should be constrained by:

A. Capabilities of selected test facilities and environments;

B. Level of testing to be accomplished; and

C. Fidelity and capabilities of existing simulations.

This page left blank intentionally.

6 SOFTWARE PRODUCT EVALUATIONS

Software Product Evaluations (SPEs) are a critical part of the software development process. During early phases of the product development, the contractor will internally coordinate the product with the appropriate teams for evaluation. The objectives of each evaluation are to ensure that the products are technically adequate and to identify and correct defects. A SPE is performed on deliverable products produced during each product development phase as shown in Table 6-1 and required by DOD-STD-2167A, as well as, DOD-STD-2168 as defined in the following paragraphs.

TABLE 6-1 SOFTWARE PRODUCTS PHASES AND OCCURRENCE

PRODUCT	PHASE						
	Sys Req Anal/ Design	SW Req Anal	Prelim Design	Detail Design	Code & CSU Test	CSC Int & Test	CSCI Test
Software Development Plan	X						
Software Requirements Specification		X					
Software Test Plan			X				
Software Test Description (Test Cases)				X			
Source Code					X	X	
Software Product Specification			X	X			X
Version Description Document							X
Software Users Manual							X
Firmware Support Manual							X
Software Programmers Manual							X
Interface Control Documents (Prime)		X Part 1	X Part 2				

DOD-STD-2168 is the standard for software quality assurance activity, which the S&MA organization follows, and is performed in concert with the IPTs to provide an assurance of the products' quality. The Prime SQA role in the SPE is defined in the SQPP volume of the Prime S&MA Plan. The intent is for the SPEs to be performed in parallel with the software development, and be fully integrated with the other activities of the contract, so that problems can be detected and corrected early. The team approach is used to perform SPEs so that available team talent is used and elimination of any duplication between the IPT and S&MA activity is achieved.

The method for accomplishing the SPE is described below:

A. For all formally deliverable products, SPEs, as described in Section 6.2.1, will be conducted. Additional deliverable documents as identified in Table 6-1 will also be subjected to a SPE.

B. For all non-deliverable items (SDFs, informal test procedures and SIRDs), SPEs need not be conducted, however a quality evaluation will be accomplished by representatives of SQA and appropriate IPTs, as deemed necessary.

During the product development phase, walkthroughs may be performed several times. Only after SPE discrepancies have been resolved, along with a follow-up SPE, if deemed necessary by the SPE leader, will the product be rendered complete and used in subsequent activities. Note that during the product's final evaluation, it is not the intent to force evaluators to repeat evaluations that have already occurred. Those conducting the final evaluation may use the results and reports from prior evaluations to verify compliance with requirements. Figure 6-1, Software Product Evaluation Process, graphically shows where SPEs occur during product development.

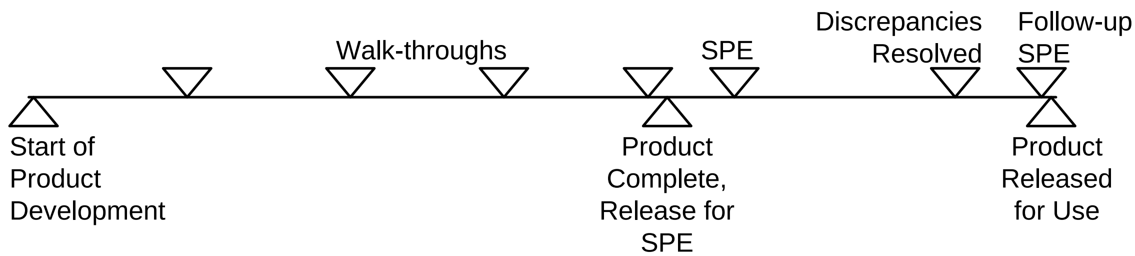


FIGURE 6-1 SOFTWARE PRODUCT EVALUATION PROCESS

6.1 ORGANIZATION AND RESOURCES - SOFTWARE PRODUCT EVALUATIONS

6.1.1 ORGANIZATIONAL STRUCTURE - SOFTWARE PRODUCT EVALUATIONS

Each software IPT will be responsible for planning and conducting SPEs on their respective products. The teams will draw upon the resources of team members and subcontractors in order to plan and conduct SPEs.

Representatives from the following teams participate, as appropriate, in SPEs:

A. Product team developing the software;

B. Those contractor teams whose disciplines cover the entire assembly element (e.g., System Safety, SCM, Test & Evaluation, and SQA); and

C. Government representative, including IV&V, when appropriate.

System Safety is represented for all Safety Critical software. These disciplines are required so that various evaluation viewpoints are covered. Each SPE member will have authority and responsibility to review the product against all criteria for that product. Any discrepancies discovered during an SPE are corrected by the IPT that created the product.

A SPE team consists of a team leader, product author, and other technical experts not responsible for developing the product under review. The team leader and technical experts are appointed by the IPT Manager. The responsibilities of the team leader include coordinating/scheduling evaluations, distributing review material, and preparing the SPE record(s). The responsibilities of the product author include preparing the product and resolving any errors/defects. The responsibilities of the technical experts include reviewing the products against the evaluation criteria and identifying errors/defects. SPE membership will be determined before the evaluation takes place, giving members adequate time for preparation. Ideally, the team should be composed of three to five people.

6.1.2 PERSONNEL - SOFTWARE PRODUCT EVALUATIONS

SPE personnel and their qualifications vary widely, depending on the product to be evaluated. Thus, there will not be a standalone SPE team. Personnel required for SPEs will be gathered on an as needed basis by the SPE team leader assigned to the respective CSCI. To ensure value added, SPE participants will have prior experience in the area being reviewed. This process is accomplished through the normal IPT structure and working relationship.

6.2 SOFTWARE PRODUCT EVALUATIONS PROCEDURES AND TOOLS

6.2.1 PROCEDURES

The SPE for Prime products is described in the following subsections.

6.2.1.1 PLANNING

During planning, the SPE team leader, in conjunction with the software IPT Manager, will:

A. Identify the specific review team for the product;

B. Schedule meetings and identify a meeting place; and

C. Distribute materials to team members, allowing adequate time for preparation.

6.2.1.2 OVERVIEW

The product author will conduct an overview session for the SPE team when requested by the SPE team leader.

6.2.1.3 PREPARATION

In preparation for the SPE review meeting, each member of the SPE team will examine the product from his or her specialty (e.g., System Safety, Test and Evaluation) point of view and also by using the criteria defined in DOD-STD-2167A. Each product has a specific set of criteria to be used in the SPE.

6.2.1.4 EXAMINATION

The team members individually review the software product, evaluating its condition relative to the evaluation criteria defined in DOD-STD-2167A, Appendix D. Collectively, the team identifies error/defects in the product. Specifically, the SPE team will perform the following tasks:

A. Examine the software product under review and verify it complies with evaluation criteria of DOD-STD-2167A. Deviations from specifications and standards will be recorded;

B. Document technical issues, related recommendations, and individuals responsible for resolving issues; and

C. Identify other issues that must be addressed.

After the software product has been reviewed, a record is generated to document the results, list deficiencies found in the product, and describe any recommendations.

When deficiencies are sufficiently critical or numerous, the SPE team leader will recommend an additional SPE review to be applied to the reworked software product after deficiencies are resolved.

6.2.1.5 COMPLETION CRITERIA

A SPE review is complete when:

A. All issues identified in the review have been addressed; and

B. SPE record has been prepared and filed.

6.2.2 TOOLS

No specific tools, other than those used for software development, are required to perform SPEs.

6.3 SUBCONTRACTOR PRODUCTS

IPT Managers are responsible for the accomplishment of SPEs for their applicable products. Subcontractor products are an integrated part of the IPT approach. Each IPT is responsible for implementing the SPE process requirements established in this document.

The Tier 1 Subcontractors should document any variance between the Prime SPE process described in Section 6.2 and their individual SPE process in the Tier 1 Subcontractor SDPs. In addition, the SQA role of the Tier 1 Subcontractors should be documented in the Tier 1 Subcontractors SQPP.

6.4 SOFTWARE PRODUCT EVALUATION RECORDS

SPE records are to be kept in the SDF with other records for CSCI products. The format of the evaluation records is up to the Tier 1 Subcontractors. The records should contain the following minimum content:

A. Evaluation date;

B. Evaluation participants;

C. Evaluation criteria;

D. Evaluation results including detected problems; and

E. Recommended correction action.

6.5 ACTIVITY - DEPENDENT PRODUCT EVALUATIONS

SPEs are conducted on each formal product developed during each software activity phase in accordance with requirements in DOD-STD-2167A and its tailoring as described in this SDP. Deliverable products will be evaluated at least once and may be evaluated more often as required. If an incremental block release process is used, updated products from each block will be evaluated during the appropriate activity phase of the block development.

This page left blank intentionally.

7 SOFTWARE CONFIGURATION MANAGEMENT

ISS CM requirements, to which software is a part, is governed by SSP 41170. Software configuration control is the responsibility of CM in support of IPTs with software requirements and development responsibilities. This includes support to the various Program Configuration Control Boards (CCBs), the software change process, the software library control process, the software audit process, the software configuration identification process, the software status accounting process, and liaison with other elements of CM.

The detailed explanation of the CM processes and methods to be used by the Prime on the U.S. segments of the ISS is contained in the CMH. The following sections relating to SCM describe the software unique CM activities and refer to the appropriate sections in the CMH for general, overall CM information. Any unique Tier 1 Subcontractor SCM activities will be described in this section within the Tier 1 Subcontractor SDPs, and will be in compliance with the ISS configuration management requirements.

7.1 ORGANIZATION AND RESOURCES - CONFIGURATION MANAGEMENT

The Prime SCM function is tiered below the System level to lower-level IPTs. Each IPT leader with a software product is responsible for assuring that SCM plans and procedures are implemented and are in conformance with the management policies and procedures. SCM serves as the primary focal point for this activity in support of the IPT leader.

The following subparagraphs describe the organization and resources necessary for Prime SCM. Each Tier 1 Subcontractor will be required to provide similar information in their SDPs.

7.1.1 ORGANIZATIONAL STRUCTURE - CONFIGURATION MANAGEMENT

The Prime SCM is a part of the VAIT and SSAIT. SCM personnel are further assigned to IPTs which have software development, build and verification responsibility.

7.1.2 PERSONNEL - CONFIGURATION MANAGEMENT

The base staffing associated with SCM varies throughout the life-cycle and generally tracks the classic software development staffing profile. In addition, SCM in its review process role, employs SCM Specialists on a temporary basis to support major milestone reviews.

7.2 CONFIGURATION IDENTIFICATION

Configuration identification includes the identification and allocation of CSCIs, CSCs and CSUs, determination of the configuration documentation required for each CSCI, the issuance of numbers and other identifiers affixed to CSCIs, CSCs and CSUs, the release of CSCIs and their associated documentation and the establishment of software configuration baselines. Configuration identification of CSCIs, CSCs, and CSUs is explained in the SSPS.

7.2.1 DEVELOPMENTAL CONFIGURATION IDENTIFICATION

Figure 3-3 presents an overview of the program phasing relationships which will be discussed in the balance of this section. The figure divides the program documentation into functional, allocated and product baselines. Identification of developmental configuration is life-cycle phase dependent. Referring to Figure 3-3, a configuration is considered developmental while it is under the control of either the Prime or the Tier 1 Subcontractor IPTs.

The functional baseline is comprised of the System Specification, the Segment Specifications and the Inter-Segment ICDs (Part 1). The functional baseline is established upon baselining of the Space Station System Specification during the SRR process; upon approval, the document is placed under Class 1 change control by NASA. The functional baseline is augmented at subsequent life-cycle reviews as shown in the figure (e.g., the Segment Specifications are initially presented at SRR, then maintained and updated under Prime CM control, and baselined at SDR (Class 1 change control)). The functional baseline describes the system's functional, physical, and interface characteristics. The functional baseline development activity consists of architectural, functional, and performance decomposition along physical and functional lines to determine segment and stage requirements. At this stage of the life-cycle, only names are used to describe the system and segments; no project unique identification is established.

The allocated baseline is comprised of the Hardware/Software Development Specifications (Type B1 and B2), the SRS and the Intra-Segment and Software ICDs (Part 1). The allocated baseline is initially presented at SSR, maintained and updated under Prime or IPT control dependent upon the document type, and baselined following successful completion of the FCA. The allocated baseline is further decomposed along functional lines, and includes identification and allocation of hardware items from an architectural viewpoint, and then identification and allocation of CSCIs to specific hardware items. With the identification of the CSCIs, a set of functional requirements is decomposed to determine the capabilities required of the software; this activity results in creation of the SRS and Part 1 of the Software ICDs. At this stage of the life-cycle, the CSCIs are assigned a name and a program unique identifier. The rules and guidelines governing the naming conventions is defined in the Prime SSPS. The process the Prime will be using is defined in the CMH.

The software product baseline is comprised of the SPSs, the VDDs and the Software ICD (Part 2). The products that will eventually form the product baseline are initially presented at software PDR, maintained and updated under Prime or IPT control dependent upon the document type, and placed under baseline control following successful completion of the PCA. The product baseline activity is the design, development, test and evaluation of software products to meet the CSCI requirements. The software development organization presents a CSCI software architecture comprised of CSCs and CSUs to meet the intent of the allocated baseline. During the product baseline activity, the CSCs and CSUs are assigned program unique identifiers by the software development organization. The rules and guidelines governing the naming conventions and program unique identifiers for software is defined in the Prime SSPS. The naming, marking, and numbering convention process the Prime is using is defined in the CMH.

Each Tier 1 Subcontractor will decompose their CSCIs into lower-level CSCs and CSUs. For the developmental configurations identified in the above process, the contents will be governed by the DOD-STD-2167 applicable DIDs as documented in the Prime SSPS.

7.2.2 IDENTIFICATION METHODS

Configuration identification establishes baseline identification of both deliverable and non-deliverable software products and documentation. The primary instrument for baselining the software configuration (both code and documentation) is the review and approval process. The review process consists of reviews by appropriate AITs/IPTs. Non-delivered code is controlled by the developing contractor; it will be identified and put under configuration control by the developing contractor's CM organization.

CSCIs for deliverable items will be initially identified and selected by the IPTs. The final confirmation of CSCI identification for deliverable items is accomplished during the applicable software PDR. Subsequent identification of CSCs and CSUs begins at software PDR and is baselined at CDR. Each software CSCI version and each block software release will be uniquely identified and controlled. The naming, marking, and numbering convention process is identified in the CMH, and the rules and guidelines governing the naming conventions and program unique identifiers for software are defined in the Prime SSPS.

7.3 CONFIGURATION CONTROL

All proposed changes to the baseline to which the Prime has control, will be properly documented, evaluated, coordinated and dispositioned according to the procedures outlined in Appendices C & L of the CMH.

7.3.1 FLOW OF CONFIGURATION CONTROL

The IPT engineering organizations are the suppliers of the proposed configuration changes. SCM manages the collection, coordination and control of this process in direct support of the IPTs. SQA will evaluate the changes and the change control procedures which process and implement changes to assure conformance to requirements and standards.

In the Prime CM system there are two levels of control: Internal Change Control and External Change Control.

Internal Change Control is maintained at the corresponding AIT/IPT levels. Each IPT will implement a software change control system which tracks and processes changes to software and documentation. (See Section 3.10 of this plan for a description of the Corrective Action Process). Software related changes will be submitted to SCM, who will serve as the processing focal point for proposed changes. SCM will assign a number, enter it into a tracking system, and route copies for coordination. The change will be submitted to the corresponding AIT/IPT for disposition and coordination. If it is determined that the AIT/IPT has been empowered to disposition the change (i.e., it is within their control and has no other impacts), they can approve the change. SCM will complete the entries for the tracking system and maintain copies of the change package in the files.

External Change Control is identical to the Internal Change Control process when PCMs are progressing through the AIT/IPT layers. Changes which impact the contractual baseline, have external impacts, are above the contractual authority, or affect baselines maintained at a higher level will be submitted to the next higher team, until they reach the level that has authority to disposition the change. If the PCMs rise above the Prime's span of control, the government mandated change process forms will be used.

Government controlled baseline documentation will be controlled by the Class 1 change process. Internal baseline documentation that has been released, but not yet baselined by the customer, will be controlled via internal configuration control at the appropriate level (e.g., IPT or Tier 1 Subcontractor), as appropriate, to the document.

7.3.2 REPORTING DOCUMENTATION

Change documentation consists of Engineering Change Proposals (ECPs), Program Change Proposals (PCPs), Team Change Proposals (TCPs), Internal Change Forms, Specification Change Notices, Document Change Notices, Preliminary Interface Revision Notice/Interface Revision Notice and Deviations and Waivers. The Prime documentation requirements are defined in the CMH. The CMH addresses the format, contents, and instructions for completing the change documentation.

With the exception of the PCM, each of the Tier 1 Subcontractors has in place both forms and procedures compliant with their previous SSFP NASA center contract. These forms and procedures will continue to be used to the extent they are compliant with the ISS configuration management requirements.

7.3.3 REVIEW PROCEDURES

The CMH describes the purpose of and the procedures to be employed by the review boards associated with the flow of Prime Configuration Control. The Tier 1 Subcontractor unique review procedures will be described in their SDPs.

7.3.4 STORAGE, HANDLING AND DELIVERY OF PROJECT MEDIA

Section 3.9 of this plan outlines the function of the Software Support Library at each of the contractual levels. The methods and procedures to be followed by the CM members of the IPT and boards will be described in the CMH. Storage, handling and delivery of project media should be addressed in the Tier 1 Subcontractor SDPs either directly or through reference to their Configuration Management Plan, at their discretion. Figure 7.3.4-1, Storage, Handling and Delivery of Project Media, depicts the storage, handling and delivery of project media.

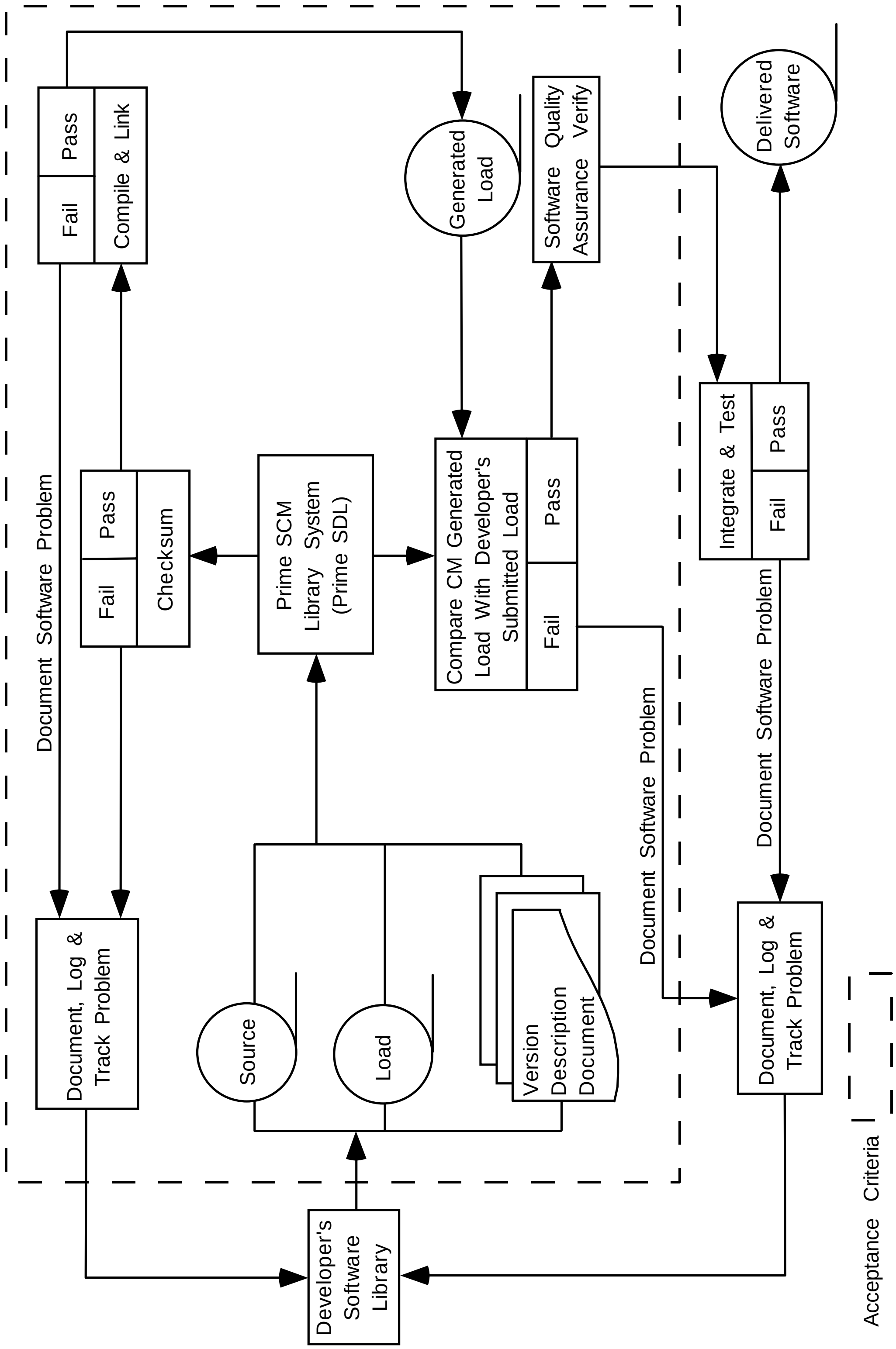


FIGURE 7.3.4-1 STORAGE, HANDLING AND DELIVERY OF PROJECT MEDIA

7.3.5 ADDITIONAL CONTROL

There are no additional Prime configuration control activities identified. The Tier 1 Subcontractors should utilize this section in their SDPs to describe any additional control activities.

7.4 CONFIGURATION STATUS ACCOUNTING

Configuration status accounting provides the information needed to identify the configuration and determine the status of change proposals and deviations and waivers. The status accounting report for a CSCI provides information concerning traceability of configuration baselines and changes to them, computer programs, specific CSCIs and their lower level components, and related documentation in a controlled access environment. The technique for tracking and reporting the configuration status of CSCIs for the Prime is the same as that for hardware, and will be tracked in the Change and Commitment Tracking Information System.

Configuration status accounting records and reports assure that there will be a configuration record documenting all approved configuration changes to all CSCIs. Periodic status reports will be provided on all products in the developmental configuration and the allocated and product baselines. These reports have the following objectives:

A. Provide traceability of changes to controlled products;

B. Serve as a basis for communicating the status of each CSCI; and

C. Serve as a vehicle for ensuring that delivered documents describe and represent the associated software.

Two reports that should be maintained by the Prime configuration status accounting are the Configuration Identification Index (CII) and the Configuration Status Accounting Report (CSAR). These records document CSCI status from initial design acceptance until program completion; thus a current account of approved CSCIs, approved changes to CSCIs, and actual versus approved configuration is maintained. Software status accounting also tracks the status of software changes and enhancements. This information will be entered into the CII and CSAR, where appropriate, when this information is maintained separately.

The SCM IPT will review configuration records to verify an accurate accounting of configuration status. Prime status accounting will include, at a minimum:

A. Current change schedule;

B. Change progress indication;

C. Summary of current changes including:

    1. Reason for change

    2. Impact of change;

D. Summary of current SPR status; and

E. Definition of key milestones and deliverables.

Section 3.12 presents additional software metric information which will be collected as a function of the software life-cycle phase.

Configuration status accounting of software items by the Prime will be accomplished as specified in the CMH. Configuration status accounting should be addressed in the Tier 1 Subcontractor SDPs either directly, or through reference to their Configuration Management Plan at their discretion.

7.5 CONFIGURATION AUDITS

As part of the software life-cycle, audits of software are scheduled between CDR and delivery. A FCA/PCA of each CSCI is held following completion of either FQT or End Item HW/SW I&V. Where it is cost effective, several CSCIs or CSCs may be audited concurrently. Both CM and SQA participate in audits to assure they are in conformance with requirements.

In addition to the formal reviews and audits specified in Section 3.8 of this SDP, the Prime SCM IPT will conduct configuration audits on Tier 1 Subcontractor software products. How the Prime will conduct these audits is identified in the CMH. Each Tier 1 Subcontractor will define the process for conducting audits on their lower-level subcontractors in their SDPs.

7.6 PREPARATION FOR SPECIFICATION AUTHENTICATION

Software specifications (SRS and SPS) for the ISS program will be authenticated by the Prime as defined in the CMH. Specification changes after authentication by the Prime will be documented and processed as described in the CMH.

7.7 CONFIGURATION MANAGEMENT MAJOR MILESTONES

Software products will be evaluated at the milestone reviews specified in Section 3.8 of this plan. Configuration baseline establishment associated with the successful completion of these milestones by the Prime is defined in the CMH.

This page left blank intentionally.

8 OTHER SOFTWARE DEVELOPMENT FUNCTIONS

8.1 PATCH/PPL PROCESS

This section describes the process for development of software patches to code and adaptation data for a CSCI that has completed FCA/PCA. Figure 8.1-1, Patch/PPL Process Flow, depicts the Patch/PPL process. Detailed instances of this process are documented in D684-10293-01, Software Configuration Management Handbook.

The Patch/PPL Process is a streamlined version of the Development Process and accommodates quick turnaround. The streamlined nature of the Patch/PPL Process is partly obtained by making adjustments to the nominal TRR and Acceptance Review (AR). The key points of the process are identified in Figure 8.1-1,

Rev B

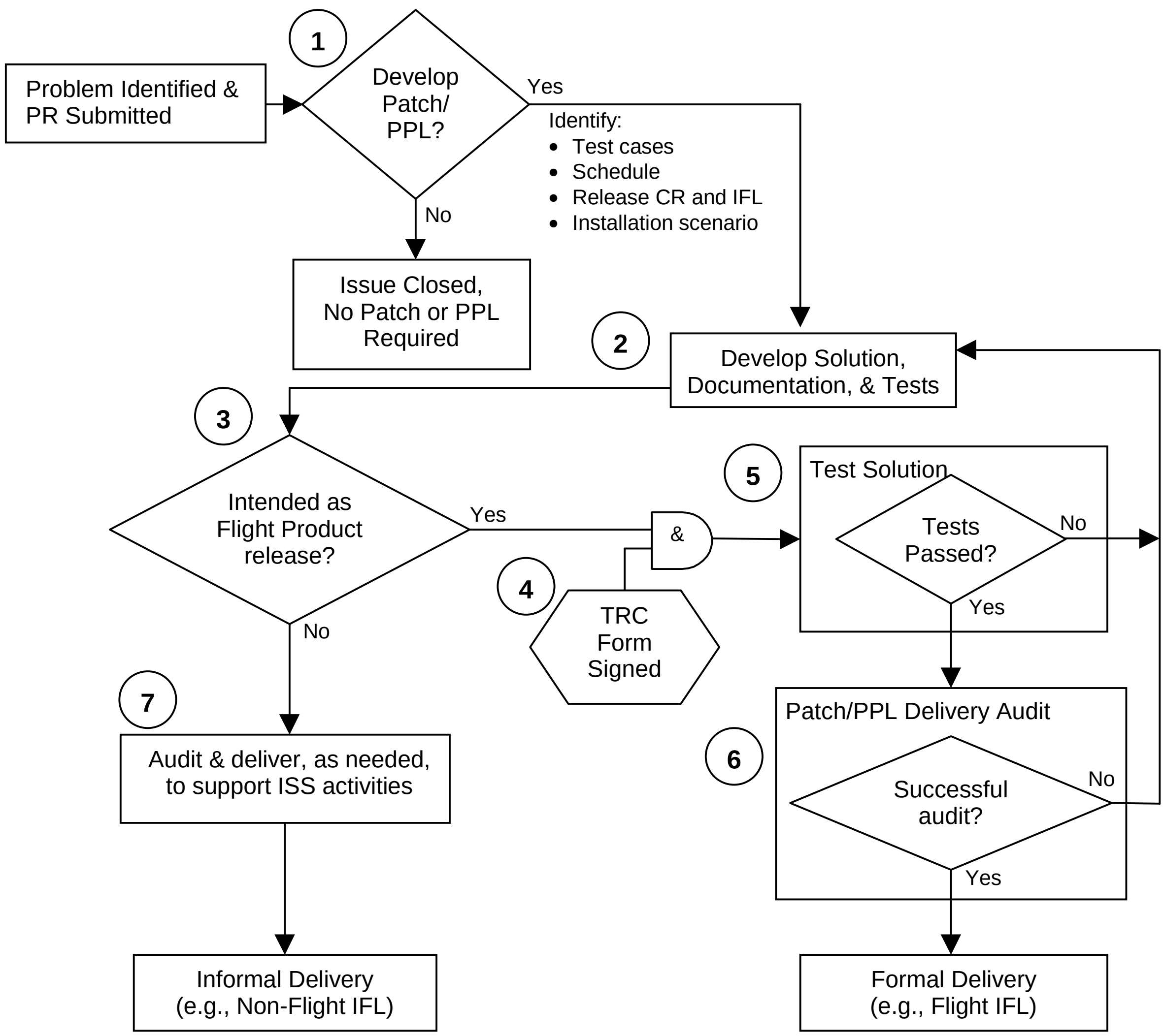


FIGURE 8.1-1 PATCH/PPL PROCESS FLOW

1) Decide if a Patch/PPL will be developed.

Data patches and PPLs are used to routinely change adaptation data. Code patches are typically reserved for high-priority, simple changes. Factors such as problem severity, code complexity, size, and permanence are considered for code patches. Once a decision has been made to develop a Patch/PPL, other key decisions can be made. Test cases to verify the Patch/PPL are selected. The IFL and CR to be used for release of the Patch/PPL are identified, and the Patch/PPL is scheduled. The installation scenario for the Patch/PPL is also approved. If the schedule need for the Patch/PPL is routine, then the normal scheduling and review process for a Problem Report (PR) is followed and the normal panels and working groups are used for decision making. When quick-turnaround is required, the normal scheduling and review process is bypassed and these decisions are made in a joint working group comprised of key members of the software panels.

To facilitate completion of a 24 hour turn around a Patch/PPL Expeditor will be appointed by Program Management at the time it is decided a quick turn around is required. The Patch/PPL Expeditor is responsible to Program Management for the successful completion of the quick turn around. This coordinator will own the Test Readiness Concurrence check-list and Acceptance Criteria check-list and is responsible for coordinating the obtaining all necessary signatures. Rev B

2) Develop the Solution.

If a PPL or a patch to an Adaptation Data Table (ADT) is being developed, the software designer provides a PPL or ADT Requirement Data File (RDF) to the subsystem data owner to obtain redlines of the new table values. The RDF is a configuration-controlled document that defines the engineering values for all instances/versions of a particular PPL or ADT. The software designer and the subsystem data owner specify the format of the RDF. The software designer builds the RDF, and submits the signed RDF to Software CM. The designer and the subsystem data owner jointly develop the PDD content updates. The software designer develops the VDD and LIF for the Patch/PPL and tests the LIF before delivering the VDD, PDD, LIF, and RDF to Software CM. SCM, with SQA in the Flight Patch Build Process, verifies the VDD, PDD, LIF, and RDF. Software development folders are maintained throughout the process. Rev B

For code patches, the software designer identifies any un-used code within the CSCI as a result of the patch. Boeing and NASA safety organizations review the un-used code for risk of inadvertant activation while on-orbit. The software designer and safety organization will work together until the risk is acceptable for use on the vehicle.

An IFL working group prepares the IFL Build Spec and Software CM builds the IFL based on the Build Spec. Software CM places the IFL in the Mission Build Facility after being validated by SQA. Software CM also makes the RDF available to Mission Control Center-Houtson (MCC-H). The software tester modifies the test procedures/scripts and develops success criteria for each test by working with the subsystem data or requirement owner. The software lab maintenance team sets the test rig to the test configuration and Software QA certifies the test rig.

Rev B

3) Flight Maturity Product or Not

Patch/PPLs will not be used on the ISS vehicle until all qualification and certification for flight has been completed. However, many product deliveries are intended to support ground activities such as hardware/software integration testing, verification test dry-runs, etc. These products will not be used for flight use until qualified and therefore are not subject to the rigorous auditing standards imposed for flight certified products. Non-flight maturity products will be audited as documented in the SCM Handbook for delivery to the ISS program for non-flight use.

4) Obtain Test Readiness Concurrence.

The Patch/PPL Process does not contain a formal Test Readiness Review. To ensure all required items are ready for testing, a Test Readiness Concurrence (TRC) sign-off sheet must be completed for all PPLs and patches requiring testing as part of their qualification. ALL code patches must be verified by testing. PPLs and other adaptatiohn data patches may be qualified by other means, as specified in the RDF. Table 8.1-1, Patch/PPL Test Readiness Checklist, defines those items that must be signed-off as being ready for testing. The Patch/PPL Test Readiness Concurrence sign-off sheet used to capture signatures is provided in the SCM Handbook. Boeing SQA will ensure all signatures on the TRC, which is pre-requisite to the start of testing.

TABLE 8.1-1 PATCH/PPL TEST READINESS CHECKLIST

Items to be verified
PPL values/ patch design
RDF under CM Control (PPL/ADT only)
Test case selection
Test success criteria
FQT/Stage Certified test rig(s)
VDD, PDD, and LIF ready for test
Configuration control of VDD, PDD, LIF (IFL ready if stage test)

5) Test the Patch/PPL.

If FQT testing is required for the Patch/PPL, the FQT Tester obtains the software files from Software CM. If Stage testing is required, the software tester obtains the software files from the IFL. When both FQT and stage tests are to be performed they may be run concurrently, but stage testing may not be signed off until FQT is signed off. Tests are run using the test procedures and are witnessed by SQA including NASA SQA and Defense Contract Management Command (DCMC). Problems that are found in testing are re-worked by the software designer until the test is successfully completed. After successful test completion, the software tester delivers the Test Preparation Sheet (TPS), formal test procedure, test report, and test results to Software QA.

If qualification of the PPL or adaptation data patch is to be done via analysis or other means, then this block is the conduct of that analysis or other qualification method. Rev B

6) Audit the Patch/PPL prior to delivery of a Flight Maturity Level Product.

A delivery audit is conducted to ensure that all the required steps of the Patch/PPL process have been performed and that the Patch/PPL is ready for delivery. Table 8.1-2, Patch/PPL Delivery Audit Completion Criteria, lists all the items to be audited. Boeing SQA ensures a completely signed Delivery Audit Completion Form prior to formal delivery. If a quick turn-around is required, Boeing SQA coordinates a single meeting of key software representatives to discuss issues, sign the Delivery Audit Completion Form, and approve the CR. Problems that are found in the delivery audit are re-worked by the participants until a fully staged Delivery Audit Completion Form is obtained. A fully signed Patch/PPL Delivery Audit Completion Form is a pre-requisite formal delivery of a Patch/PPL. Boeing SQA coordinates formal delivery of the Patch/PPL with SCM and Software Engineering based on contractual requirements and also whether delivery occurs pre- or post-DD250 of the basic CSCI that contains the functionality that is being patched. After final CR approval and formal delivery, Software CM makes the IFL in the MBF available to the final end-user of the Patch/PPL only after SQA signs the IFL VDD.

7) Audit and Delivery of a non-Flight Maturity Level Product.

Audit and delivery of a non-Flight Maturity Level Product is conducted per the proceses defined within the SCM Handbook.

TABLE 8.1-2 PATCH/PPL DELIVERY AUDIT COMPLETION CRITERIA

Patch Type	Items to be examined for accuracy and completeness
Code	By-passed code within CSCI, as a result of patch, has been identified and is acceptable
All	Software Development Folder complete
All	Completed Test Readiness Concurrence form; test rig software matches IFL Build Spec (single CSCI and associated Patches/PPL's for FQT rig)
All	IFL completed and matches IFL spec; PDD Patch IFL and Compatible Software IFL correct
All	Successful test completion with TPS, formal test procedure, test report, results delivered
All	PDD purpose, installation, constraints, subsystem, function affected, and stage applicability correct
All	PDD command and telemetry effects correct
All	PDD checksums and/or filenames, MDM configuration and constraints match test results
All	PDD CSCI version and patches, PR number are correct
PPL, ADT	PDD RDF number and version correct if PPL or ADT

Rev B

8.2 SUSTAINING ENGINEERING CSCI RELEASE PROCESS

The Sustaining Engineering CSCI Release Process deals with changes to a CSCI where a new CSCI executable must be created and released. This process does not deal with those changes to a CSCI handled as a patch (see Section 8.1). Figure 8.2-1, Sustaining Engineering CSCI Release Process, provides a pictorial representation of this process.

1. Initiate change

The need to release a new CSCI executable can come about from

a) Resolving problems documented on one or more Problem Reports;
b) The need to implement a new capability (requires a Program Change); or
c) By direction of NASA.

2. Determine course of action

Problem Reports are first handled by the Joint Software Review Panel (JSRP) or Data Integration Team (DIT) and then passed onto the ASCB. Program Changes and NASA direction are handled directly by the ASCB, where upon concurrance a Problem Report is written. The ASCB will determine whether or not to procede with a new release of the CSCI.

3. Define content and develop draft schedule

Once ASCB has given direction for a new release of the CSCI , the developer will provide a draft Schedule Issue Form (SIF)/schedule to the SRCP based on the required delivery date of the new release. The SRCP will coordinate the release schedule with all users and produce a Proposed SIF.

4. Develop final content and schedule

The developer and NASA will work together to define the content of the Program Change. The developer will provide the SRCP a final proposed schedule with a list of PRs/ Station Program Notes (SPN)s making up the release. As additional PRs are identified by the JSRP requiring software correction they may identify this sustaining release as the required release. These PRs will be taken to the SRCP for inclusion in the SIF and for negotiation of any schedule impacts.

5. Finalize schedule and reviews

The Proposed SIF will be presented to ASCB for approval and direction to implement changes.

6. Develop solution

Upon obtaining authorization, the developer will develop the change. The develoers will hold the design reviews as needed. The Test Readiness Review will always be held. See Section 8.2.1 for more information on the Software Development Processs.

7. Release product

Rev B

Once the change is developed and tested the CSCI can only be delivered through the Audit/Delivery process via a Program Change that authorizes the delivery.

Rev B

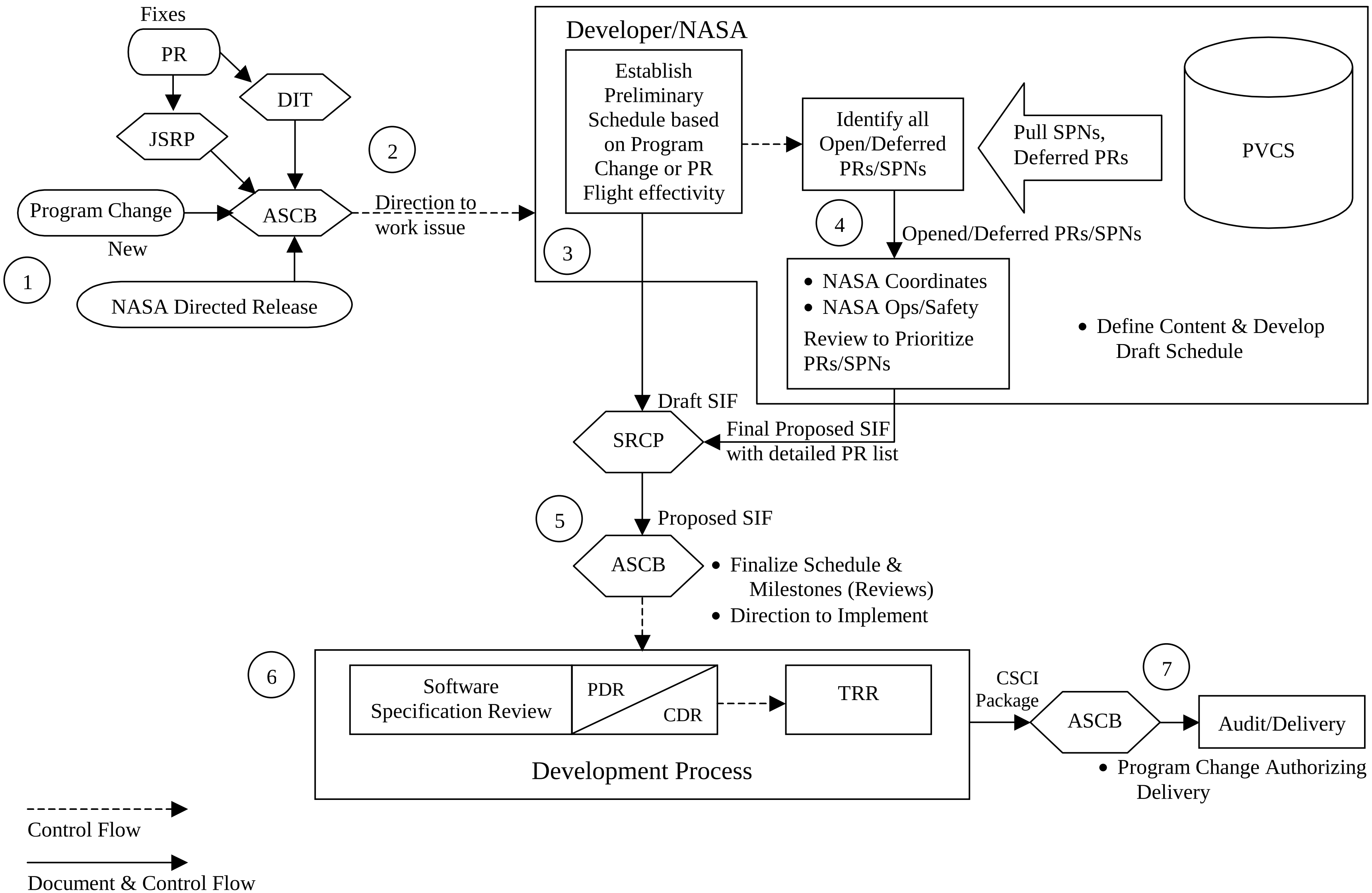


FIGURE 8.2-1 SUSTAINING ENGINEERING CSCI RELEASE PROCESS

Rev B

8.2.1 SOFTWARE DEVELOPMENT PROCESS

This software development process is tailored from that used during full development of the CSCI. The only review that must take place is the TRR with the design reviews occuring as directed by ASCB or where reviewable documents for those reviews have changed.

The result of this software development process will be an update to the baseline of the CSCI in question.

8.2.1.1 SOFTWARE DEVELOPMENT PROCESS REVIEWS

The developer is responsible for coordination and tracking of issue generation and resolution of issues resulting from the following reviews of their products. The status and progress of issue tracking will be provided to the ASCB as requested and at specified developer meetings.

Prior to closure of issues, the issue Actionee must coordinate the proposed resolution with the issue originator to obtain agreement on the resolution. Any issue where consensus on the resolution cannot be reached is elevated to the ASCB to be worked. Records of closure of the issues will be maintained in the corresponding SDF.

The reviews discussed below are conducted as team member reviews. The design reviews (SSR, PDR, & CDR) to be held may, at the discretion of the developer, be held simultaneously based on the expected design development work.

Reviews will be held for NASA by the developer as documented in this SDP. The developer will provide the status of specific review results and recommendation for review disposition to ASCB. These reviews may take various forms, but will meet the intent of the following standard MIL-STD-1521B reviews:

A. Software Specification Review (SSR);

B. Preliminary Design Review (PDR);

C. Critical Design Review (CDR); and

D. Test Readiness Review (TRR).

8.2.1.1.1 SOFTWARE SPECIFICATION REVIEW

The purpose of the SSR is to demonstrate to NASA the adequacy of the software and interface requirements to proceed into the design phase.

A SSR will be held if and only if any of the SSR reviewable documents change or as directed by the ASCB.

Inputs to and definition of successful completion for SSR are per those given in Section 3.8.2.1, Software Specification Review, with modifications for successful completion as listed here:

Rev B

A. Acceptance by review community of the new and changed requirements as stated in the SRS; and

K. Action plan for SSR issue resolution and definition of new and changed functional requirements and external interfaces to 95% completion by PDR including assignment of actionee and closure data for each issue.

8.2.1.1.2 PRELIMINARY DESIGN REVIEW

The purpose of the PDR is to determine if the top-level design of the software is mature and complete enough to advance to the detailed design phase.

A PDR will be held if and only if any of the PDR reviewable documents change or as directed by the ASCB.

Inputs to and definition of successful completion for PDR are per those given in Section 3.8.2.2, Preliminary Design Review, with modifications for successful completion as listed here:

A. Acceptance by review community of the requirements as stated in the SRS and confirmation that these constitute approximately 95% of the total of new and changed requirements (i.e., issues and action plans are less than 5%); and

H. Approximately 80% of new and changed detailed external interface data defined with action plans for completing remaining interface definition.

8.2.1.1.3 CRITICAL DESIGN REVIEW

The purpose of the CDR is to determine if the detailed design of the software is correct, consistent and complete enough for development to continue to coding and informal testing. This technical review is held to provide a detailed basis for verifying design integrity and compatibility with CSCI requirements and assessment of formal test preparation.

A CDR will be held if and only if any of the CDR reviewable documents change or as directed by the ASCB.

Inputs to and definition of successful completion for CDR are per those given in Section 3.8.2.3, Critical Design Review, with modifications for successful completion as listed here:

A. Acceptance by review community of the requirements as stated in the SRS and confirmation that these constitute approximately 98% of the total of new and changed requirements (i.e., issues and action plans are less than 2%); and

B. Acceptance by review community of software design as stated in the SSP/SDF and confirmation that these constitute approximately 95% complete of the new and changed design.

8.2.1.1.4 TEST READINESS REVIEW

Rev B

The purpose of the TRR is to ensure that the software test procedures are complete and carry out the intent of the software test plan and descriptions and software to be tested is under formal control and ready for test. This review will be conducted after software test procedures are available and CSC integration testing has been successfully completed.

A TRR will always be held.

Inputs to and definition of successful completion for TRR are per those given in Section 3.8.2.4, Test Readiness Review.

9 NOTES

9.1 ACRONYM LIST

AC	Assembly Complete
ACBSP	Assembly Contingency Baseband Signal Processor
ACRFG	Assembly Contingency Radio Frequency Group
ADO	Adaptation Data Overlay
ADP	Acceptance Data Package
ADT	Adaptation Data Table
AG	Application Generator
AIS	Automated Information Systems
AIT	Analysis and Integration Team
AIUA	Audio Interface Unit Audio
AIUC	Audio Interface Unit Controller
AL	Airlock
ALSYS	Airlock System
AMP	Audio Management Processor
APM	Attached Pressurized Module
AR	Acceptance Review
ASCB	Avionics Software Control Board
ATP	Acceptance Test Procedure
ATU	Audio Terminal Unit
B-CP	Boeing-Canoga Park
B-HB	Boeing-Huntington Beach
B-HOU	Boeing-Houston
B-HSV	Boeing-Huntsville
BSDG	Boeing Software Development Group
BMP	Bus Management Processor
C&C	Command and Control
C&DH	Command and Data Handling
CASE	Computer Aided Software Engineering
CBMBC	Common Berthing Mechanism Bolt Controller
CBMMLC	Common Berthing Mechanism Master Latch Controller
CCB	Configuration Control Board
CCC	Control Center Complex
CCS	Command and Control Software
CDL	Central Data Library
CDR	Critical Design Review
CFE	Contractor Furnished Equipment
CFEL	Contractor Furnished Equipment List
CGS	Columbus Ground System
CI	Configuration Item
CII	Configuration Identification Index

Rev B

Rev B

CM	Configuration Management
CMG	Control Moment Gyro
CMH	Configuration Management Handbook
CoFR	Certificate of Flight Readiness
COTS	Commercial Off-The-Shelf
CPL	Central Program Library
CPU	Central Processing Unit
CR	Change Request
CSAR	Configuration Status Accounting Report
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSDL	Charles Stark Draper Laboratories
CSL	Central Software Library
CSU	Computer Software Unit
DCMC	Defense Contract Management Command
DDCU	DC/DC Converter Unit FC
DID	Data Item Description
DIL	Deliverable Items List
DIT	Data Integration Team
DPI	Data Processing Installation
DR	Discrepancy Report, Data Requirement
DRAM	Dynamic Random Access Memory
DRD	Data Requirements Description
DRL	Data Requirements List
DVO	Detailed Verification Objective
ECP	Engineering Change Proposal
ECU BGA	Electronic Control Unit Beta Gimbal Assembly
ECU SAW	Electronic Control Unit Solar Array Wing
EMDM	Enhanced MDM
EMDMBF	Enhanced MDM Boot and Diagnostics Firmware
EMP	Engineering Management Plan
EEPROM	Electrically Erasable Programmable Read Only Memory
EPROM	Erasable Programmable Read Only Memory
ESA	European Space Agency
ETCS	External Thermal Control System
EVSU	External Video Switch Unit
EXT	External
FC	Firmware Controller
FCA	Functional Configuration Audit
FE	Factory Equipment
FEU	Functionally Equivalent Unit
FGB	Functional Cargo Block
FQT	Formal Qualification Test

Rev B

FSE	Flight Software Environment
FSM	Firmware Support Manual
FSSR	Flight System Software Requirements
FSW	Flight Software
GFE	Government Furnished Equipment
GFEL	Government Furnished Equipment List
GN&C	Guidance Navigation and Control
GPS	Global Positioning System
GSE	Ground Support Equipment
HW	Hardware
HI	Honeywell Incorporated
HRDL	High Rate Data Link
HSYS	Habitation System
HWCI	Hardware Configuration Item
IACO	Integration and Check Out
ICD	Interface Control Document
ICP	Interface Control Plan
ICWG	Interface Control Working Group
IFL	Integrated Flight Load
IFQT	Incremental FQT
IMCA	Integrated Motor Controller Assembly
INTSYS	Internal System
I/O	Input/Output
IP	International Partner
IP&CL	Instrumentation Programming and Command List
IPT	Integrated Product Team
ISCDR	Integrated Stage Critical Design Review
ISFQR	Integrated Stage Formal Qualification Review
ISL	Integrated Signal List
ISPDR	Integrated Stage Preliminary Design Review
ISS	International Space Station
ISSSR	Integrated Stage Software Specification Review
ISTRR	Integrated Stage Test Readiness Review
ITRR	Incremental TRR
I&V	Integration and Verification
IV&V	Independent Verification and Validation
JEM	Japanese Experiment Module
JSC	Johnson Space Center
JSRP	Joint Software Review Panel
KB	Kilobytes
KhSC	Khrunichev State Scientific Production Space Center

Rev B

KSC	Kennedy Space Center
LDI	Local Data Interface
LIF	Load Image File
LP&S	Lab Power and Switching
LSYS	Laboratory System
MATE	MDM Application Test Environment
MBF	Mission Build Facility
MCA	Major Constituent Analyzer
MCC-H	Mission Control Center-Houston
MDM	Multiplexer/Demultiplexer
MOTS	Modified Off the Shelf
MOU	Memorandum of Understanding
MPLM	Mini Pressurized Logistics Module
MPS	Module Pump System
MSS	Mobile Servicing System
N2SYS	Node 2 System
NASA	National Aeronautics and Space Administration
NCS	Node 1 Control Software
ND	Non Deliverable
NDS	Non-Developmental Software
NLT	No Later Than
NR	Not Required
OIU	Orbiter Interface Unit
ORD	Operational Readiness Date
ORU	Orbital Replacement Unit
PALS	Program Automated Library System
PC	Personal Computer
PCA	Physical Configuration Audit, Pressure Control Assembly
PCM	Program Change Memo
PCP	Program Change Proposal
PCS	Portable Computer System
PCU	Plasma Contactor Unit
PDD	Patch/PPL Description Data
PDL	Program Design Language
PDR	Preliminary Design Review
PEP	Program Execution Plan, Payload Executive Processor
PG	Product Group
PIDS	Prime Item Development Specification
PMA	Power Management Assembly
PMCA	Power Management and Control Application
PMCU	Power Management Control Unit

Rev B

PMI&VP	Program Master Integration and Verification Plan	
PPL	Pre-Positioned Load	
PPMC	Pump Package Motor Controller	
PR	Problem Report	
PROM	Programmable Read Only Memory	
PTR	Port Thermal Radiator	
PUI	Program Unique Identifier	
PVCA	Photovoltaic Controller Application	
PVCU	Photovoltaic Controller Unit	
PVIS	Program Verification Information System	
RAM	Random Access Memory	
RDF	Requirements Data File	Rev B
RDMA	Risk Data Management Application	
RF	Radio Frequency	
RG	Rate Group	
RPCM	Remote Power Control Module	
ROM	Read Only Memory	
RS	Russian Segment	
RSA	Russian Space Agency	
RTE	Run-Time Environment	
S0	Starboard 0	Rev B
S&MA	Safety and Mission Assurance	
SARJ	Solar Array Rotary Joint	
SCA	Switchgear Controller Assembly	
SCM	Software Configuration Management	
SCU	Sync and Control Unit	
SDD	Software Design Document	
SDF	Software Development Folder	
SDIL	Software Development Integration Lab	
SDL	Software Development Library	
SDMS	Structure Dynamic Measurement System	
SDP	Software Development Plan	
SDR	System Design Review, Software Discrepancy Report	
SDRL	Subcontractor Data Requirements List	
SDS	Supplier Data Sheet	
SEE	Software Engineering Environment	
SG TRC	Space to Ground Transmitter/Receiver Controller	
SGS HRFM	Space to Ground to Space High Rate Frame MUX	
SGS HRM	Space to Ground to Space High Rate Model	
SIF	Schedule Issue Form	Rev B
SIRD	Software Implementation Requirements Document	
SLOC	Software Lines Of Code	
SM&C	Station Management & Control	
SMDM	Standard MDM	

SMDMBF	Standard MDM Boot and Diagnostic Firmware
SOW	Statement of Work
SPD	Serial/Parallel Data
SPE	Software Product Evaluation
SPM	Software Programmer's Manual
SPN	Station Program Note
SPR	Software Problem Report
SPS	Software Product Specification
SQA	Software Quality Assurance
SQPP	Software Quality Program Plan
SRB	Software Review Board
SRR	System Requirements Review
SRS	Software Requirements Specification
SSAIT	Space Station Analysis and Integration Team
SSFP	Space Station Freedom Program
SSP	Space Station Program
SSPS	Software Standards and Procedures Specification
SSR	Software Specification Review
STD	Software Test Description
STP	Software Test Plan
STR	Software Test Report, Starboard Thermal Radiator
SUM	Software User's Manual
SVF	Software Verification Facility
SW	Software
SWI	Software Integration
TBD	To Be Determined
TCMS	Test, Control and Monitor System
TCP	Team Change Proposal
TDCS	Test Data Capture Software
TEP	Team Execution Plan
TIM	Technical Interchange Meeting
TP	Tone Processor
TPS	Test Preparation Sheet
TRC	Test Readiness Concurrence
TRR	Test Readiness Review
TSE	Test Support Equipment
UHF	Ultrahigh Frequency
UIL	User Interface Language
UPA	Urine Processor Assembly
U.S.	United States
USGS	United Stated Ground Segment
USOS	United States On-orbit Segment
VAIT	Vehicle Analysis and Integration Team

Rev B

VBSP	Video Baseband Signal Processor
VDD	Version Description Document/Drawing
VLN	Verification Logic Network
VTS	Video Test Set
WP	Water Processor
XPDR	Transponder

9.2 GLOSSARY

Brassboard Software

A highly structured software product of a higher quality than the prototype software, produced prior to the completion of the design phase to provide greater confidence that the software design will meet the system functional and performance requirements. This software product differs from throw-away prototype code in that it will be upgraded to become final product code.

Catastrophic/Critical/High Risk Software

Software where a failure to execute or an inappropriate execution results in:

Damage that will result in loss of mission; delay of launch; or severe hazard to the crew.

Central Software Library

The central electronic repository of flight software and data which have been delivered to the MBF for integration, test, and incorporation into one or more mission flight builds.

Code Complete

Follows completion of CSU testing. (Proof of product code and test driver maturity to perform CSU testing is assured by a peer review TRR.)

Code Upgrade

Process by which brassboard software is upgraded based on results of Software Development Test Article and program standards to become product code.

Computer Program

A sequence of coded instructions which encode a thought process or algorithm that may be executed by a computer system.

Data Processing Installation

A grouping of one or more automated information systems based on common function, facility, management, and/or logic. The DPI includes the host computer(s), the operating system software, locally attached workstations, internal networks, external network interfaces, and the physical environment within which the systems reside.

Deliverable Software

Software, including flight software, ground support software, support and development software, simulation software, and test software, that is to be delivered to the Prime.

Developmental Software

All software not defined as reusable software, commercially-available software, or government-furnished software.

Early Functional Integration

The step within the Software Development Test Article phase that integrates key software subfunctions to assure that a CSCI will meet its functional requirements.

Facility Software

Software required to support, control, or run a test facility.

Flight Software

Body of operational software on the ISS at a given point in time and applies to all flight software CSCIs developed for use with FCs, MDMs, sensors, effectors, and other ISS defined flight ORUs. For qualification and acceptance tests of hardware using flight software, the software may be modified to the extent necessary to conduct the qualification and acceptance tests.

Formal Qualification Test

Process that allows the Prime to determine whether a software configuration item complies with the allocated requirements for that item.

Grandfathering

The process for determining which SSFP software products can be reused for the ISS program. This provides for the use of existing standards, processes and documentation and establishes the point in the software life-cycle where ISS development starts.

Ground Software

Software used to support on-orbit mission and includes crew and payload training, CCC operations, analytical tools, and models.

Ground Support Equipment

Contract-deliverable equipment (hardware/software) used on the ground to test, transport, access, handle, maintain, measure, verify, service, and protect flight hardware/software.

Ground Support Equipment Software

Software that is contained in GSE.

Integrated Flight Load (IFL)

A collection of ISS Program software and data that functions together to accomplish specific purposes such as for a flight or a particular test.

Integrated Flight Load Build Specification (IFL Build Spec)

Is the definition of a particular Integrated Flight Load as well as defining the configuration of each component included in the Integrated Flight Load.

Module

The lowest level software design/implementation unit (i.e., Unit or CSU, a subdivision of a CSC).

Non-deliverable Software

Software that is developed and used within a Tier 1 Subcontractor and is not being delivered to the Prime.

Non-developmental Software

Includes reusable software, commercially-available software, and government-furnished software.

Non-flight Software

Software used to support ground activities including ISS flight software design, development, integration, and verification; flight article and end-item design, development, and qualification; on-orbit stage configuration integration and verification; and launch package integration. Categories include test software, including simulations; SVF software; GSE/TSE software; and ground software, including MBF software.

Operational Increment

The release of a CSCI configuration designed and developed to provide a defined set of functionality in support of a specific station configuration, test, or other program need.

Platform Services vs. MDM Executive

Platform services are those provided to the application software which provide the interface between the target platform and the application. The MDM Executive is that portion of the application software which sequences and controls the execution of the application.

Ported Software

Software which is transferred from one platform/environment to another (i.e., software which is developed on (or for) a platform different from the targeted platform). The software is ported from the development platform to the target platform.

Prototype Software

Initial software developed to determine feasibility of software design and provide initial proof of system requirements. Explorations in prototype software will provide feedback to requirements changes. For the ISS program, prototype software will be developed for all high program risk, and critical functions.

Requirements Analysis

Software Requirements Analysis is a specific term which applies to specific processes which are performed after the software requirements are defined and is the key to an effective software design. It is used in both the Structured Analysis and Object-Oriented development approaches.

Reusable Software

Software that is reusable which provides low-risk, low-cost to meeting ISS software requirements. Reusable Software must meet all process requirements and all data items associated with this software must meet format and content standards for the ISS software.

Simulation

Provides the environment for the integration and verification of the flight software and avionics from software development through integration. Simulations include the test environment with end item and segment simulations, environmental simulations which represent the ISS on-orbit environment and dynamics, and sensor/effector simulations.

Software Test Article Development

Early software development process using Brassboard software to assure that the software product will meet system functional requirements. This process feeds the Simulation/Flight Software Integration and Flight Software Code Development phases.

Software Product Evaluations

Evaluations that are performed on deliverable products produced during each product development phase to ensure compliance to requirements.

Software Verification Facility

An ISS program facility comprised of multiple test strings of functionally equivalent MDMs connected with test unique hardware and software that will support flight software stage verification and validation and operational procedure validation.

Standard Output Interface Definition

This specification defines and controls at a requirements level, the MBF output products necessary to support reconfiguring the ISS and NASA ground facilities. Standard Output is the electronic delivery mechanism for all USOS signals and flight software.

Support Software

Categories of support software include facility, test, GSE/TSE and ground software.

Test Software

Types of test software include vertical simulations, horizontal simulations, test software, and test configured flight software.

Test Support Equipment

Equipment that is designed for use by the contractor to support development, production, and test activities associated with the ISS flight hardware and software.

Test Support Equipment Software

Software that is contained in TSE.

Unused Software

Software which is not executed as part of the CSCI in which it resides (e.g., an Ada Run-Time Environment (RTE) capability not used by the specific CSCI).

Validation

Process of ensuring that what is intended to be built corresponds to what is actually required; it is concerned with the completeness, consistency and correctness of the requirements.

Verification

Process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase.

APPENDIX A CSCI LIST

This appendix contains tables of CSCI products for the U.S. portion of the ISS program, by software developer. All of the information is based on the program understanding at the time of the document release. Table A-1, SVF Software, lists the SVF CSCIs. Table A-2, Flight Software CSCI and Major CSC, contains the Flight Software CSCIs and shared CSCs. Shared CSCs are those produced by a Tier 1 Subcontractor other than the CSCI Owner. Table A-3, Ground Software including MBF, lists the Ground Software CSCIs, including MBF, Table A-4, GSE/TSE Software, lists the GSE/TSE software CSCIs, and Table A-5, Test Software Including Simulations, lists the test software, including flight software simulations for use in FQT.

TABLE A-1 SVF SOFTWARE

Rev B

CSCI Name	Provider	Capability
SVF Output CSCIs		
Base Support	B-HOU	Non-real time support
Real Time Process Control Support Services	B-HOU	Non-real time support
Real Time Display Support Services	B-HOU	Non-real time support
Post Processing Support Services	B-HOU	Non-real time support
Session Setup and Staging Support Services	B-HOU	Non-real time support
TDCS Capture	B-HOU	Test data capture
TDCS Retrieval	B-HOU	Test data capture
TDCS Data Acquisition	B-HOU	Test data capture
SVF Input CSCIs		
CES Command and Control Environment Simulation	B-HOU	Simulation
NES Node 1 Control Environment Simulation	B-HOU	Simulation
ETCS Simulation	B-HB	Simulation
GN&C Simulation	B-HB	Simulation
EXT Simulation	B-HB	Simulation
SARJ Simulation	B-HB	Simulation
SDMS Simulation	B-HB	Simulation
Secondary Electrical Power System Simulation	B-CP	Simulation
Power Management Control Application MATE Simulation	B-CP	Simulation
Photovoltaic Control Application MATE Simulation	B-CP	Simulation
Integrated Lab Simulation through 8A	B-HSV	Simulation
Integrated Lab, Node2 & Hab Simulation through 15A	B-HSV	Simulation
Integrated Lab, Node2 & Hab Simulation through AC	B-HSV	Simulation
Lab Power & Switching	B-HOU	Simulation

TABLE A-2 FLIGHT SOFTWARE CSCI AND MAJOR CSC

Location	CSCI Acronym	1st Flt	CSCI/ CSC	CSCI Name	Devel- oper	Host MDM	Lang Used
FGB	FGB	1A	CSCI	Functional Cargo Block Control	KhSC	FGB-1,-2	Ada
			CSC	SMDM Utilities	B-HB/HI	FGB-1,-2	Ada/ Asm
FGB, S1, Node 1, Lab, P1, Node 2, Hab, AL, S3, P3, S4, P4, S6, P6, PMA1, S0	SMDMBF	1A	CSCI	Standard MDM Boot and Diagnostics FC	B-HB/HI	PVCU-1-4A, PVCU-1-4B, S0-1,-2, S1-1,-2, P1-1,-2, S3-1,-2, P3-1,-2, HAB-1,-2,-3, N1-1,-2, AL-1, STR-1, FGB-1,-2, PTR-1, N2-1,-2, LA-1,-2,-3	Ada/ Asm
FGB, S1, Node 1, Lab, P1, Node 2, Hab, AL, S3, P3, S4, P4, S6, P6, PMA1, S0	SPD 1553	1A	CSCI	Serial/Parallel Data Card 1553 Firmware	B-HB/HI	All MDMs	Asm
PMA1	NCS	2A	CSCI	Node 1 Control Software (Release 1)	B-HOU	N1-1,-2	Ada
			CSC	SMDM Utilities	B-HB/HI	N1-1,-2	Ada/ Asm
Node 1, JEM, Hab, Lab, Node 2	CBMMLC	2A	CSCI	Common Berthing Mechanism Master/Latch Controller	B-HSV	FC	C
Node 1, JEM, Hab, Lab, Node 2	CBMBC	2A	CSCI	Common Berthing Mechanism Bolt Controller	B-HSV	FC	C

Rev B

TABLE A-2 FLIGHT SOFTWARE CSCI AND MAJOR CSC (Continued)

Location	CSCI Acronym	1st Flt	CSCI/ CSC	CSCI Name	Devel- oper	Host MDM	Lang Used
Node 1, AL, Lab, Z1, S1, P1, Node 2, S0, Hab, S3/S4, P3/P4	RPCM	2A	CSCI	Sec. Power Control Application FC	B-CP	FC	Asm
PMA1	NCS R2	2A.1	CSCI	Node 1 Control Software (Release 2)	B-HOU	N1-1,-2	Ada
			CSC	SMDM Utilities	B-HB/HI	N1-1,-2	Ada/ Asm
P1, S1	ACBSP	3A	CSCI	S-Band Baseband Signal Processor FC	B-HB	FC	C
P1, S1	ACRFG	3A	CSCI	S-Band RF Group FC	B-HB	FC	Ada
P1, S1	XPDR	3A	CSCI	Standard TDRSS Transponder FC	B-HB	FC	C
Z1	SG TRC	3A	CSCI	Ku-Band Transmitter/Receive r Controller FC	B-HB	FC	Ada
Z1	PCU	3A	CSCI	Plasma Contactor Unit FC	B-CP	FC	Asm
P4, S4, P6, S6	PVCA	4A	CSCI	Photovoltaic Controller Application	B-CP	PVCU-1-4A, PVCU-1-4B	Ada
			CSC	SMDM Utilities	B-HB/HI	PVCU-1-4A, PVCU-1-4B	Ada/ Asm
PV	ECU BGA	4A	CSCI	Electronic Controller Unit/Beta Gimbal Assembly FC	B-CP	FC	Asm
PV	ECU SAW	4A	CSCI	Electronic Controller Unit/Solar Array Wing FC	B-CP	FC	Asm
PV, S0	LDI	4A	CSCI	Local Data Interface FC	B-CP	FC	Asm

Rev B

TABLE A-2 FLIGHT SOFTWARE CSCI AND MAJOR CSC (Continued)

Location	CSCI Acronym	1st Flt	CSCI/ CSC	CSCI Name	Devel- oper	Host MDM	Lang Used
Lab, S0, S1, P1, Node 2, Hab	DDCU	4A	CSCI	DC/DC Converter Unit FC	B-CP	FC	Asm
S0	SCA	4A	CSCI	Switch Gear Controller Assembly FC	B-CP	FC	Asm
Lab, S0	EMDMBF	5A	CSCI	Enhanced MDM Boot and Diagnostics FC	B-HB/HI	C&C-1,-2,-3, GN&C-1,-2, PL-1,-2, INT-1,-2, EXT-1,-2, PMCU-1,-2	Ada/ Asm
Lab, S0	HRDL	5A	CSCI	High Rate Data Link Firmware	B-HB/HI	C&C-1,-2,-3, GN&C-1,-2, PL-1,-2, INT-1,-2, EXT-1,-2, PMCU-1,-2	Ada/ Asm
Lab, AL, Hab	PCA	5A	CSCI	Pressure Control Assembly	B-HSV	FC	Ada/ Asm
Lab	GN&C	5A	CSCI	GN&C MDM	B-HB	GN&C-1,-2	Ada
			CSC	EMDM Utilities	B-HB/HI	GN&C-1,-2	Ada/ Asm
Lab	SGS HRFM	5A	CSCI	Ku-Band High Rate Frame MUX FC	B-HB	FC	C
Lab	SGS HRM	5A	CSCI	Ku-Band High Rate Modem FC	B-HB	FC	C
Lab, Hab, Node 2	PPMC	5A	CSCI	Pump Package Motor Controller	B-HSV	FC	C
Lab	VBSP	5A	CSCI	Ku-Band Video Baseband Signal Processor FC	B-HB	FC	Ada
Lab	CCS	5A	CSCI	Command & Control Software	B-HOU	C&C-1,-2,-3	Ada
			CSC	EMDM Utilities	B-HB/HI	C&C-1,-2,-3	Ada/ Asm
			CSC	Timeliner UIL Kernel and Adapter	CSDL*	C&C-1,-2,-3	

Rev B

TABLE A-2 FLIGHT SOFTWARE CSCI AND MAJOR CSC (Continued)

Location	CSCI Acronym	1st Flt	CSCI/ CSC	CSCI Name	Devel-oper	Host MDM	Lang Used
Lab	PMCA	5A	CSCI	Manage Electric Power Systems	B-CP	PMCU-1,-2	Ada
			CSC	EMDM Utilities	B-HB/HI	PMCU-1,-2	Ada/ Asm
			CSC	Control Solar Alpha Rotary Joint	B-HB	PMCU-1,-2	Ada
Lab	LSYS1	5A	CSCI	LAB Systems 1	B-HSV	LA-1	Ada
			CSC	SMDM Utilities	B-HB/HI	LA-1	Ada/ Asm
Lab	LSYS2	5A	CSCI	LAB Systems 2	B-HSV	LA-2	Ada
			CSC	SMDM Utilities	B-HB/HI	LA-2	Ada/ Asm
Lab	LSYS3	5A	CSCI	LAB Systems 3	B-HSV	LA-3	Ada
			CSC	SMDM Utilities	B-HB/HI	LA-3	Ada/ Asm
Lab	PEP	5A	CSCI	Payload Executive Processor	B-HSV	PL-1,-2	Ada
			CSC	EMDM Utilities	B-HB/HI	PL-1,-2	Ada/ Asm
			CSC	Timeliner UIL Kernel and Adapter	CSDL*	PL-1,-2	
LAB	INTSYS	5A	CSCI	Internal Systems	B-HSV	INT-1,-2	Ada
			CSC	EMDM Utilities	B-HB/HI	INT-1,-2	Ada/ Asm
			CSC	SEPS - Control and Monitor RPCMs	B-CP	INT-1,-2	Ada
Lab, Hab	AMP	5A	CSCI	Audio Management Processor	B-HSV	FC	Ada/ Asm
Lab, AL, Node 2, JEM, APM, Hab	ATU	5A	CSCI	Audio Terminal Unit	B-HSV	FC	C / Asm
Lab, Hab	AIUA	5A	CSCI	Audio Interface Unit Audio	B-HSV	FC	C / Asm
Lab	TP	5A	CSCI	Tone Processor	B-HSV	FC	C / Asm
Lab, Hab	BMP	5A	CSCI	Bus Management Processor	B-HSV	FC	C / Asm

Rev B

TABLE A-2 FLIGHT SOFTWARE CSCI AND MAJOR CSC (Continued)

Location	CSCI Acronym	1st Flt	CSCI/ CSC	CSCI Name	Devel-oper	Host MDM	Lang Used
Z1	CMG	5A	CSCI	CMG Control FC	B-HB	FC	Asm
Lab, Hab	AIUC	5A	CSCI	Audio Interface Unit Command	B-HSV	FC	C / Asm
Lab	SCU	5A	CSCI	Sync and Control Unit	B-HSV	FC	C / Asm
Lab, Hab	MCA	6A	CSCI	Major Constituent Analyzer	B-HSV	FC	Ada/ Asm
Lab	HCOR	6A	CSCI	High Rate Communications Outage Recorder	B-HB/ SEAKR	FC	Ada
Airlock	ALSYS1	7A	CSCI	Airlock Systems 1	B-HSV	AL-1	Ada
			CSC	SMDM Utilities	B-HB/HI	AL-1	Ada/ Asm
Node 2, S0, Lab	EVSU	8A	CSCI	External Video Switch FC	B-HB	FC	Ada
S0	EXT	8A	CSCI	External MDM (Release 1)	B-HB	EXT-1,-2	Ada
			CSC	EMDM Utilities	B-HB/HI	EXT-1,-2	Ada/ Asm
			CSC	SEPS - Control and Monitor RPCMs	B-CP	EXT-1,-2	Ada
P3/P4, S0, S3/S4, S1, P1	IMCA	8A	CSCI	Integrated Motor Controller Assy FC	B-HB	FC	Asm
S0	RG	8A	CSCI	Rate Gyro FC	B-HB	FC	Asm
S0	EXT R2	9A	CSCI	External MDM (Release 2 - Adding TRRJ, EATCS & Complete DSM)	B-HB	EXT-1,-2	Ada
			CSC	EMDM Utilities	B-HB/HI	EXT-1,-2	Ada/ Asm
			CSC	SEPS - Control and Monitor RPCMs	B-CP	EXT-1,-2	Ada
S1, P1	MPS	9A	CSCI	Pump Module Assembly FC	B-HB	FC	Ada / C
S1, P1	S1/P1	9A	CSCI	S1/P1 MDM	B-HB	S1-1,-2, P1-1,-2	Ada

Rev B

TABLE A-2 FLIGHT SOFTWARE CSCI AND MAJOR CSC (Continued)

Location	CSCI Acronym	1st Flt	CSCI/ CSC	CSCI Name	Devel-oper	Host MDM	Lang Used
			CSC	SMDM Utilities	B-HB/HI	S1-1,-2, P1-1,-2	Ada/ Asm
S1, P1	STR/PTR	9A	CSCI	STR/PTR MDM	B-HB	STR-1,-2, PTR-1,-2	Ada
				SMDM Utilities	B-HB/HI	STR-1,-2, PTR-1,-2	Ada/ Asm
Node 2	N2SYS1	10A	CSCI	Node 2 Systems 1	B-HSV	N2-1	Ada
			CSC	SMDM Utilities	B-HB/HI	N2-1	Ada/ Asm
Node 2	N2SYS2	10A	CSCI	Node 2 Systems 2	B-HSV	N2-2	Ada
			CSC	SMDM Utilities	B-HB/HI	N2-2	Ada/ Asm
S0	S0	10A	CSCI	S0 MDM	B-HB	S0-1,-2	Ada
			CSC	SMDM Utilities	B-HB/HI	S0-1,-2	Ada/ Asm
S0	EXT R3	12A	CSCI	External MDM (Release 3 - Adding SARJ & CAS)	B-HB	EXT-1,-2	Ada
			CSC	EMDM Utilities	B-HB/HI	EXT-1,-2	Ada/ Asm
			CSC	SEPS - Control and Monitor RPCMs	B-CP	EXT-1,-2	Ada
S3, P3	S3/P3	12A	CSCI	S3/P3 MDM	B-HB	S3-1,-2, P3-1,-2	Ada
			CSC	SMDM Utilities	B-HB/HI	S3-1,-2, P3-1,-2	Ada/ Asm
Hab	UPA	13A	CSCI	Urine Processor Assembly	B-HSV	FC	Ada
Hab	HSYS1	15A	CSCI	HAB Systems 1	B-HSV	HA-1	Ada
			CSC	SMDM Utilities	B-HB/HI	HA-1	Ada/ Asm
Hab	HSYS2	15A	CSCI	HAB Systems 2	B-HSV	HA-2	Ada
			CSC	SMDM Utilities	B-HB/HI	HA-2	Ada/ Asm
Hab	HSYS3	15A	CSCI	HAB Systems 3	B-HSV	HA-3	Ada
			CSC	SMDM Utilities	B-HB/HI	HA-3	Ada/ Asm

Rev B

TABLE A-2 FLIGHT SOFTWARE CSCI AND MAJOR CSC (Concluded)

Location	CSCI Acronym	1st Flt	CSCI/ CSC	CSCI Name	Devel-oper	Host MDM	Lang Used
Hab	WP	16A	CSCI	Water Processor	B-HSV	FC	C
Any module with an MDM	MDMLU†	n/a	CSCI	MDM Loader Utility	B-HB/HI	Any MDM	Ada
Lab	SSMMU	TBD	CSCI	Solid State Mass Memory Unit	B-HB/HI	Lab	Ada
* GFE † The MDMLU is uploaced upon demand							

Rev B

TABLE A-3 GROUND SOFTWARE INCLUDING MBF

CSCI/CSC	Software Developer	Destination
Data Integration Tool Set	B-HB	MBF
Mission Database Application	ESA	MBF
Oracle RDBMS	Oracle	MBF
Software Processing & Storage Tool	B-HB	MBF
Software Release Distribution Tool (SRDT)	B-HB	MBF
SQL *Net	Oracle	MBF
SQL *Pro Ada	Oracle	MBF
SQL *Pro C	Oracle	MBF
Sun Ada	SUN	MBF
SunOS Operating System	SUN	MBF
VAX Ada	DEC	MBF
VAX C	DEC	MBF
VaxSet Software Management Tools	DEC	MBF
VMS Operating System	DEC	MBF
Simulations I/O Services	B-HB	FEUs
EEPROM Header Tool	B-HB	MDMs
Timeliner Compiler Kernel and Adapter	GFE	MBF
Aonix Alsys AdaWorld VAX/VMS to 80386 Cross Compiler w/Optimizer, Debugger, Linker	Aonix	MBF
Aonix Alsys AdaWorld VAX/VMS to 80386 Problem Reporting Compiler	Aonix	MBF
Aonix ActivAda Real-Time Cross to 80386 Problem Reporting Compiler (Windows NT Platform)	Aonix	MBF
Destination - e.g. Segment, LAB, ORU, SVF, etc.		

Rev B

TABLE A-4 GSE/TSE SOFTWARE

CSCI Acronym	CSCI /CSC	CSCI Name	SW Developer
VTS	CSCI	Video Test Set	B-HB
	CSCI	Ku-Band Test Set	B-HB
	CSCI	S-Band Test Set	B-HB

Rev B

TABLE A-5 FQT SUPPORT SOFTWARE INCLUDING SIMULATIONS

CSCI Name	SW Developer	Destination
HSYS1SIM Hab Systems 1 Simulation	B-HSV	MATE
HSYS2SIM Hab Systems 2 Simulation	B-HSV	MATE
HSYS3SIM Hab Systems 3 Simulation	B-HSV	MATE
LSYS1SIM Lab Systems 1 Simulation	B-HSV	MATE
LSYS2SIM Lab Systems 2 Simulation	B-HSV	MATE
LSYS3SIM Lab Systems 3 Simulation	B-HSV	MATE
N2SYS1SIM Node 2 Systems 1 Simulation	B-HSV	MATE
N2SYS2SIM Node 2 Systems 2 Simulation	B-HSV	MATE
PCASIM Pressure Control Assembly Simulation	B-HSV	AG
WPSIM Water Processor Simulation	B-HSV	AG
ALSYS1SIM Airlock Systems 1 Simulation	B-HSV	MATE
PEPSIM Payload Executive Processor Simulation	B-HSV	MATE
INTSYSSIM Internal Systems Simulation	B-HSV	MATE
CES Command and Control Environment Simulation	B-HOU	MATE
NES Node 1 Control Environment Simulation	B-HOU	MATE
ETCS Simulation	B-HB	MATE
GN&C Simulation	B-HB	MATE
External Simulation	B-HB	MATE
SARJ Simulation	B-HB	MATE
SDMS Simulation	B-HB	MATE
IACO MATE Simulation Components	B-HB	MATE
Power Management Control Application MATE Simulation	B-CP	MATE
Photovoltaic Control Application MATE Simulation	B-CP	MATE
Local Bus Controller	B-HB	PC
Sensor / Effector Simulator	B-HB	TBD
MATE-3 Development and Control Services	B-HB	MATE
MATE-3 I/O Services	B-HB	MATE
MATE-3 I/O Utilities	B-HB	MATE

Rev B

This page left blank intentionally.

APPENDIX B SOFTWARE METRICS

The following guidelines are provided to assist the Space Station software development community in the implementation of software metric reporting. Software metrics are comprised of the following:

- Software size;
- Design, code and informal test progress;
- Software volatility;
- Formal test progress; and
- Computer resource utilization.

Each metric is addressed separately in the following paragraphs, including a description, reporting period, and format.

B.1   GENERAL

The following data should be reported quarterly for each CSCI and major subsystem CSC as defined in Appendix A:

- CSCI/CSC name;
- CSCI/CSC manager (Software Development Manager) and phone number;
- Tier 1 IPT or applicable subcontractor name;
- Reporting period (as-of-date will be the last Friday of the calendar month);
- System/element name; and
- Target processor identification.

The reporting period for metrics is calendar quarter, i.e., the data reported on October 15, 1994 reflects data for the quarter of July through September 1994, etc. All reporting should be in spreadsheet format and should be transmitted electronically to the software metric coordinator by close of business on the 15th of the month following the end of the quarter.

Data which is not applicable to a CSCI/CSC for a reporting period should be left blank, (e.g., integration and test progress would not be reported until design and code are complete). If the actual data for a specific metric has not changed from the previous quarter, the previous quarter’s data should be repeated.

In addition to the estimated and actual data, a separate report to include planning data, as detailed in the descriptions below, should be provided for each CSCI/CSC at the start of each development phase. The metric categories involved are: design, code and test progress, and formal test definition and execution.

B.2 SOFTWARE SIZE

B.2.1 DESCRIPTION

This metric allows program management to track changes in the effort required to complete each project and to estimate the effort required to support the resulting products. Software size is the primary input parameter to most software estimation and utilization models. An increase in software size can lead to schedule slips and staffing inadequacies. A trend of increasing software size should result in corrective actions that either counter the trend or accommodate it with increased effort during the remaining development phases.

B.2.2 UNITS OF MEASURE

Software size will be reported as SLOC by CSCI/CSC for each language used. Four categories of SLOC will be reported by language, newly developed code, existing code that is modified, existing code that is reused as-is, and COTS code. Code should be separated into modules that contain only one category of SLOC: new, modified, reused (both ported and as is), or COTS.

B.2.3 INFORMATION REPORTING

Software size is reported for each language separately. For example, if a CSCI/CSC is made up of both Ada-language code and C-language code, there will be two reports for that CSCI/CSC. One report will consist of counts for new, modified, reused (both ported and as is) and COTS SLOC for the Ada-language portion of the CSCI/CSC, and the second report will have counts for new, modified, reused and COTS SLOC for the C-language portion. For Ada, a SLOC is counted as a non literal semicolon. For all other languages, a SLOC is counted as any line of program text that is not a comment or blank line, regardless of the number of statements or fragments of statements on the line.

A SLOC may include the following:

- Each source statement created by project personnel and processed into machine code;
- Job control language;
- Format statements;
- Data declarations; and
- Newly developed support software (i.e., debug statements).

SLOC will not include comments. SLOC count will include all code developed for and included in the processor except for the MDM Boot and Diagnostics software (e.g. the CCS CSCI includes SLOC counts for Timeliner).

Reported SLOC counts will include the following:

- Total estimated SLOC to be developed by language in each category for the completed CSCI/CSC; and
- Actual SLOC completed in each category during the reporting period.

Actual SLOC counts will be reported as the code is placed under CM control in preparation for the start of FQT and should be cumulative through the reporting period. If the COTS software is to be modified, the counting rules described herein apply, with the unmodified software being reported as "COTS."

Macro expansions or SLOC supplied to the program (e.g., by a "with" or "include" statement) should be counted only once for all modules being measured. SLOC that invoke, call, or direct inclusion of other SLOC in the module should be counted each time they are used. A "with" or "include" statement counts as one SLOC. The included file is counted separately, and only once regardless of the number of modules including it.

As changes against software requirements are approved, and brassboarding/prototyping efforts are completed, the total estimated SLOC should be revised to reflect any changes.

B.3 DESIGN, CODE AND INFORMAL TEST PROGRESS

B.3.1 DESCRIPTION

Design, code and informal test progress is measured by the number of modules (separately compilable units of code, e.g., Ada compilation units) under development by time. This metric provides visibility into the ability of the developer to keep the project on schedule, and to track progress between milestones so that problems can be identified prior to milestone reviews.

This metric tracks the development of modules through the software design, coding and informal testing processes.

B.3.2 UNITS OF MEASURE

Design, code and informal test progress is reported as module design, code and informal test completions. The completion criteria for design is successful completion of the unit design walk through. The completion criteria for coding is successful completion of a code walk through and unit testing. The completion criteria for the informal testing is successful integration into the total CSCI configuration. This integration includes all informal testing up to the point of starting FQT (i.e. includes FQT dry runs).

B.3.3 INFORMATION REPORTING

Design, code and informal test progress will be reported as follows:

- Estimated total number of modules to be developed for the complete CSCI/CSC.

- Module design completion plan to provide the number of modules/units to be designed each month for the complete design phase. This plan should be provided at the start of the phase and can be revised if required.
- Actual total number of module design completions, cumulative through the reporting period.
- Module code completion plan to provide the number of modules to be coded each month for the complete coding phase. This plan should be provided at the start of the phase and can be revised if required.
- Actual total number module code completions, cumulative through the reporting period.
- Module test completion plan to provide the number of modules to be tested each month for the complete informal testing phase. This plan should be provided at the start of the phase and can be revised if required.
- Actual total number of informal test completions.

B.4 SOFTWARE VOLATILITY

B.4.1 DESCRIPTION

Software volatility is measured by the number of changes to requirements and design (both additions and deletions) directly impacting the software development effort (i.e., issued against the internal baseline of the SRS and software after the start of formal test dry runs). It consists of the number of Change Requests (CRs) and Discrepancy Reports (DRs) or their equivalent.

B.4.2 UNITS OF MEASURE

Software volatility is reported in terms of cumulative total SLOC, CRs, and DRs. SLOCs are counted upon submittal of the code to the formal CM library prior to the start of FQT. Change Requests are counted once the developer places the SRS under internal baseline control. Discrepancy Reports are defined as the number of problems found during dry run and conduct of FQT.

B.4.3 INFORMATION REPORTING

Software volatility shall be reported for each CSCI/CSC as follows:

- Formally configuration-controlled SLOC reported as a cumulative total to date.
- Total number of CRs submitted and approved.
- Total number of CRs implemented. Implementation is considered complete when the PCR, or equivalent form, is signed off as complete.

- Total number of DRs submitted. This is a cumulative count of the number of problems written and submitted.
- Total number of new DRs opened. This is a cumulative count of problem write-ups approved by the developer's problem review board.
- Total number of DRs opened against the software. This is a cumulative count of problems approved by the developer's problem review board as requiring software changes.
- Total number of DRs closed with a software fix. The DR completion criteria is closure of the DR form with the software changes complete.

B.5 FORMAL TEST PROGRESS

B.5.1 DESCRIPTION

Formal test progress is measured as the number of formal test cases defined and executed during the test. This metric provides an indication of the ability of the contractors to meet testing schedules. Formal test progress provides data to track the progress of CSCIs, as defined in Appendix A, through all phases of formal test.

B.5.2 UNITS OF MEASURE

Formal test progress is reported in terms of test definitions and test executions. Test definition includes development of all test documentation (i.e. both volumes of the STD) in preparation for the conduct of the FQT. A test is considered defined when the procedure is placed under the tester's configuration control. The test execution count is based on the test cases as defined in the STD DOD-STD-2167A DID paragraph 10.1.6.1.1, including setup, run and shutdown and covers the period of time from the initial test run through all necessary retest and regression testing.

B.5.3 INFORMATION REPORTING

Formal test progress will be reported at the CSCI level. Separate test execution plan data for both the initial test run and any reruns will be provided. The rerun plan may not be provided until the initial test results in the need for a rerun.

The following data will be reported for each test level:

- Estimated total number of tests to be defined.
- Test definition plan to provide number of tests to be defined each month for the complete Formal testing phase. This plan should be provided at the start of the test definition phase and can be revised if required.
- Actual total number of test definitions completed to date for the CSCI.

- Test execution plan to provide number of tests to be run each month for the complete Formal testing phase. This plan should be provided at the start of the test execution phase and can be revised if required.
- Actual total number of tests run during current reporting period.
- Actual total number of tests rerun during current reporting period.
- Actual total number of tests passed during current reporting period.

B.6 COMPUTER RESOURCE UTILIZATION

B.6.1 DESCRIPTION

Computer resource utilization is a measure of Central Processing Unit (CPU) processor utilization, CPU Dynamic Random Access Memory (DRAM) utilization, CPU EEPROM utilization, and Bus utilization. This metric provides an early warning if resource capacity limits are being approached. These metrics track the projected and actual use of the target processor and I/O resources.

B.6.2 UNITS OF MEASURE

CPU processor utilization tracks the time consumed by planned or actual CSCIs/CSCs operating within the computer resource and is measured in percent utilization of CPU time for target platform.

CPU DRAM and EEPROM utilization tracks the amount of processor memory consumed by planned or actual CSCIs/CSCs allocated to the computer resource and is measured in kilobytes (KB) DRAM and EEPROM respectively.

Bus utilization tracks the MIL-STD-1553 bus resources consumed by system operations. Bus utilization is reported by each developer whose processor is the Bus Controller on either the Control, Local or User Bus. The Control and Local bus utilization is defined as the number of boxcars being used by a developer divided by the total number of boxcars allocated to that developer. Boxcars reserved for scarring will be included in both the allocated and used counts. The User Bus utilization is measured as percent utilization of available bandwidth.

B.6.3 INFORMATION REPORTING

Computer resource utilization is reported for each target processor and bus separately.

Both estimates and actuals will be reported for the complete CSCI/CSC target processor or bus, as follows:

- CPU processor utilization is reported for both the total processor and each of the five rate groups (.1 Hz, 1 Hz, 10 Hz, 80 Hz and background). Total CPU utilization will include the sum of all the rate groups and any additional tasks not captured in the rate group (e.g.

interrupts) and be reported as a percentage is measured during worst-case operational scenarios and averaged over the longest cyclic task time period. Cyclic background tasks will be counted in the background rate group. Non-cyclic background tasks which run in all unused time are not included in any rate group. CPU utilization includes the HI software.

- CPU DRAM and EEPROM utilization in KB.
- Bus utilization as a percentage, is measured during worst-case operational scenarios. The worst case scenario is averaged over a one second time period.

This page left blank intentionally.

APPENDIX C SEE TOOLS

C.1 AFFECTED ORGANIZATIONS

Table C.1-1, SEE Supported Tools, lists the SEE software tool categories which are mandated for various program participating organizations or facilities.

TABLE C.1-1 SEE SUPPORTED TOOLS

Tool Category/Name	Tier 1	MBF	SVF
Host & ADP Environment		X	
Communications		X	
Security and Data Protection		X	
Performance Assessment, Monitoring, and Management		X	
Data Base Management		X	
File Transfer		X	
Media Verification		X	
Configuration Management (Version and Change Control)		X	
Library Management Operational Software and Data		X	
Compile and Build Tools for MDM Software Production:			
Phar Lap 386I ASM Assembler	X	X	
Phar Lap 386I LINKLOC Linker	X	X	
Aonix Alsys AdaWorld VAX/VMS to 80386 Cross Development System	X	X	
Aonix Alsys AdaWorld VAX/VMS to 80386 Problem Reporting Compiler	X		
Aonix ActivAda Real-Time Cross to 80386 Problem Reporting Compiler (Windows NT Platform)	X		
Kermit	X	X	
Load Image Build Tool	X	X	X
I/O Configuration Table Build Tool	X	X	X
Displays Development and Build		X	X
Procedures Development and Build		X	X
Command Development and Build		X	X
Simulations Development and Build		X	X
Software Reuse		X	
Test Environment		X	X
Test Management		X	X
Test Results Assessment		X	X
Data Reconfiguration		X	X
MATE (*Simulations, test environment and test management)	X*		X

Rev B

This page left blank intentionally.